AD-A074 901 CALIFORNIA UNIV BERKELEY OPERATIONS RESEARCH CENTER F/G 12/2
ASPECTS OF DYNAMIC PRODUCTION PLANNING.(U)
JUL 79 R C LEACHMAN N00014-76-C-0134
UNCLASSIFIED ORC-79-8 NL

1 OF 2
AD
A074901

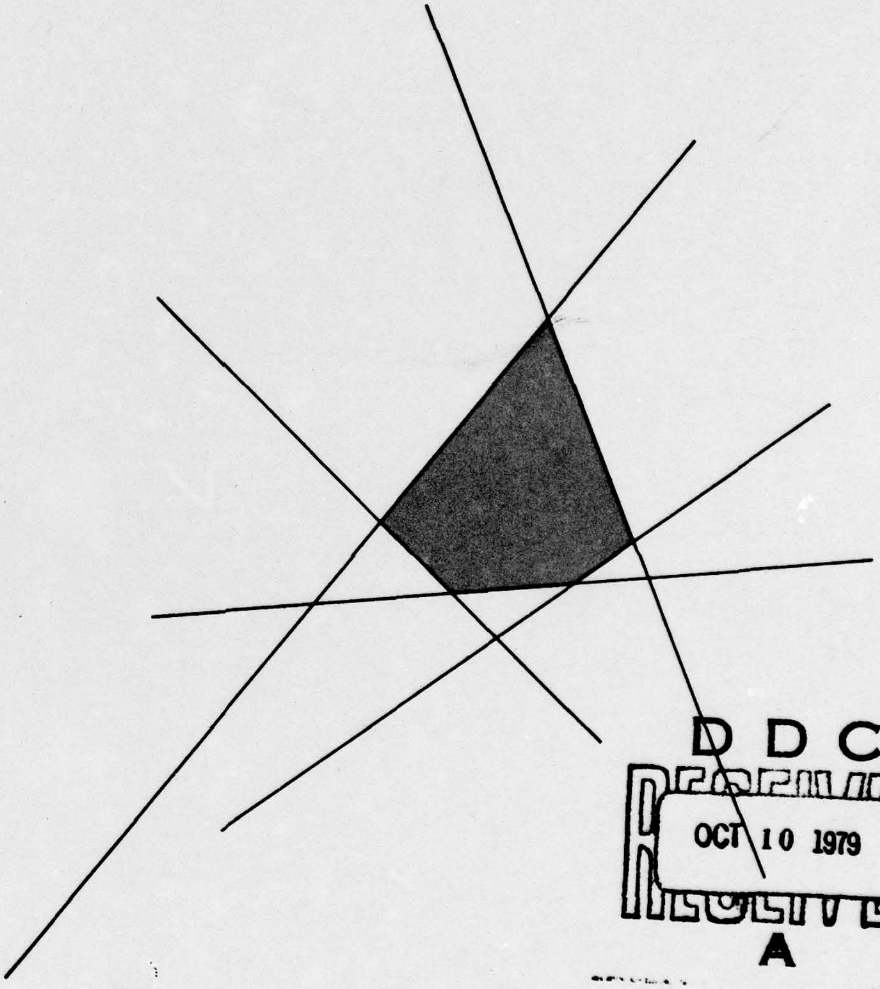
90[redacted]

ORC 79-8
JULY 1979

12

ASPECTS OF DYNAMIC PRODUCTION PLANNING

by
ROBERT C. LEACHMAN



OPERATIONS
RESEARCH
CENTER

UNIVERSITY OF CALIFORNIA · BERKELEY

ASPECTS OF DYNAMIC PRODUCTION PLANNING

by

Robert C. Leachman
Operations Research Center
University of California, Berkeley

JULY 1979 ORC 79-8

This research was supported by the Office of Naval Research under Contract N00014-76-C-0134 with the University of California. Reproduction in whole or in part is permitted for any purpose of the United States Government.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
14 ORC-79-8

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER
9

4. TITLE (and Subtitle)
6 ASPECTS OF DYNAMIC PRODUCTION PLANNING

5. TYPE OF REPORT & PERIOD COVERED
Research Report

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
10 Robert C./Leachman

8. CONTRACT OR GRANT NUMBER(s)
15 N00014-76-C-0134

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Operations Research Center
University of California
Berkeley, California 94720

12 97

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
NR 047 033

11. CONTROLLING OFFICE NAME AND ADDRESS
Office of Naval Research
Department of the Navy
Arlington, Virginia 22217

11

12. REPORT DATE
July 1979

13. NUMBER OF PAGES
97

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Production Planning
Activity Networks
Activity Intensity Histories
Resource Leveling
Capacity
Efficiency

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

(SEE ABSTRACT)

Unclassified 270 750

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

mt

ACKNOWLEDGEMENTS

The author wishes to express his debt of gratitude to Professor Ronald W. Shephard for his guidance and assistance in the research reported in this thesis.

ABSTRACT

A production system is modeled as an activity network using a dynamic activity analysis. The nodes of the network represent component activities which produce final products, intermediate products (inputs to other activities), or both, according to intensity functions varying with time. The arcs of the network indicate where transfers of intermediate product occur. These transfers of intermediate product can be modeled two ways: either as discrete units on an event basis, or as continuous flows. In the former case, a generalization of critical path planning is obtained, whereby variation in activity intensities allows variable resource load levels and production durations. The intensities provide an added degree of flexibility for production planning, smooth loading of resources, etc. In the latter case, linear programming may be used to allocate resources so as to maximize throughput, minimize production costs, and make determinations of capacity and efficiency.

Table of Contents

List of Figures

1. General Dynamic Activity Analysis

The dynamic activity analysis for production has previously been set forth in references [10] and [11], in which the structure of the dynamic activity analysis was presented as an example in the theory of dynamic production networks. This chapter is adapted from reference [10] to develop the base for concepts and procedures presented in following sections.

A production system is modelled as a network of activities which are denoted by $A_1, \ldots, A_N$. Figure 1.1 displays a typical configuration for an activity network. Each node represents an activity, which produces final products, intermediate products (inputs to other activities), or both. The arcs indicate activity dependencies, ie, where in the network transfers of intermediate product occur. In the figure, activity A_1 provides intermediate product input to A_3, while A_2 serves A_3, A_4 and A_7, etc. The double-stemmed arrows indicate activities producing final products; in this case, A_5, A_6 and A_8 all produce final output. Restrictions on the network structure appropriate to modeling certain kinds of problems will be discussed in the following sections.

In addition to the input of intermediate product, activities require exogenous inputs for operation, ie, resources not produced by other activities. The operation of each activity is allowed to vary over a discrete time grid $t=0,1,2,\ldots,$ where on each interval $[t-1,t)$, $t=1,2,3,\ldots,$ inputs (exogenous and intermediate product) are applied at uniform rates, and charged at time $(t-1)$. Outputs of production during this interval are forthcoming at time t, at which time they may be transferred as intermediate or final product, or held as part of accumulated inventories.

The operation of each activity A_i is measured in terms of an *intensity function*

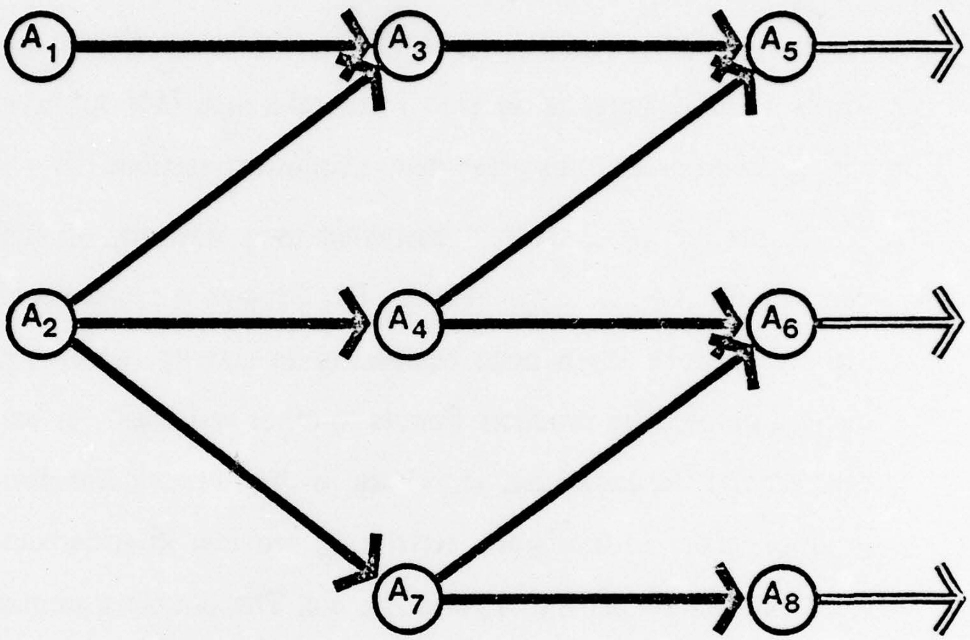


Figure 1.1
Example of Production Activity Network

$$z_i(t),\ t=0,1,2,\ldots,$$

whose value at time t, taken in conjunction with technical coefficients, indicates production input during $[t,t+1)$ and activity output at time $(t+1)$. In this fashion, activity intensity is taken to be constant on each interval $[t,t+1)$. The technical coefficients are defined as follows:

(a) $c_i(t)$, $t=0,1,2,\ldots$, $i=1,\ldots,N$, where $c_i(t)$ is the amount of output of activity A_i at time t per unit intensity of activity A_i. If the output is a sizable discrete unit, $c_i(t)$ is understood to be the fraction of a unit completed at the time t per unit intensity of operating A_i; otherwise, it represents amount (in bulk). In most applications, these output coefficients are constant in time or perhaps are direct functions of cumulative production (to indicate "learning curve" behavior), but are expressed most generally as functions of time.

(b) $a_{ik}(t)$, $t=0,1,2,\ldots$, $k=1,2,\ldots,NK$, $i=1,\ldots,N$, where $a_{ik}(t)$ is the amount of exogenous input k required at time t per unit intensity of activity A_i, and the index k spans all NK exogenous resources required by the production system. As with the output coefficients, these input coefficients are typically constant in time or perhaps direct functions of cumulative production (indicating "learning curve" behavior), but are expressed most generally as functions of time.

(c) $\bar{a}_{ij}(t)$, $t=0,1,2,\ldots$, $i=1,\ldots,N$, $j=1,\ldots,N$, where $\bar{a}_{ij}(t)$ is the amount of intermediate product from activity A_j required at time t per unit intensity of operating A_i. (In the case of discrete intermediate product transfers, constant transfer coefficients $\bar{a}_{ij}$ relate the number of units of intermediate product from activity A_j required per unit output of activity A_i. See Chapter 2.)

These coefficients are nonnegative with the following restrictions:

(a) $c_i(t) > 0$, $t = 0, 1, 2, \ldots$, $i = 1, \ldots, N$.

(b) For each $t \geqslant 0$ and each $i = 1, \ldots, N$, there exists a positive coefficient $a_{ik}(t)$ for at least one $k \in \{1, \ldots, NK\}$.

(c) For each system exogenous input $k \in \{1, \ldots, NK\}$, there exists $t \geqslant 0$ and at least one activity A_i, $i \in \{1, \ldots, N\}$, such that $a_{ik}(t)$ is positive.

The activity intensities $z_i(t)$ have natural bounds, since production cannot be carried out instantaneously. These bounds, denoted by

$$\bar{z}_i(t), \quad i=1, \ldots, N, \quad t=0, 1, 2, \ldots,$$

result from the available workspace and other limitations not considered as exogenous input. Of course, particular allocations of exogenous inputs may limit intensity further.

A *production plan* is a specification over some finite period [0,T) of the activity intensities

$$\left\{ z_i(t) \right\}_{i=1, \ldots, N}^{t=0, \ldots, T-1} .$$

Such a specification determines the production system demand histories for exogenous inputs (resources) as well as the output histories. For example, the demand for exogenous input k during the interval $[t, t+1)$ is given by

$$\sum_{i=1}^{N} z_i(t) a_{ik}(t) ;$$

and for the case that activity A_N is the only activity producing final output, and A_N produces only final product, the production system cumulative output during [0,T) is given by

$$\sum_{t=0}^{T-1} z_N(t) c_N(t+1) \ .$$

To be feasible, a production plan must of course satisfy constraints concerning intermediate product transfers, available exogenous inputs, etc., which are discussed in following sections. Expression of such constraints and further development of the model depends considerably on whether the transfers of intermediate product are considered to be large, discrete units transferred on an event basis, or are viewed as continuous flows. The former point of view seems appropriate for detailed modeling of large-scale construction such as shipbuilding, in which virtually no work on the installation of various systems can begin until large assemblies such as the hull structure are completed. On the other hand, the latter point of view is more appropriate for production processes where either output has no discrete nature or else a considerable number of discrete output units may be produced in a single time interval. Such would be the case for aggregate plants, food processing plants, paper production, and many other examples.

In the sections that follow, the model is developed first for discrete transfers of intermediate product. Starting with simple extensions of Critical Path Methods, the model is generalized into a fully dynamic structure with intensity histories, output streams, and product inventories. A transition is then made to continuous product flows for the final development of the model.

2. Dynamic Production Planning with Discrete Transfers of Intermediate Products

2.1. Nature of Discrete Product Transfers

For the discrete model of product transfers, we shall assume output of each activity is measured in discrete units, and that product transfers can only occur in integral amounts of such units. For each activity A_i, product transfers from every activity A_j supplying A_i are required for A_i to initiate production of each output unit, according to product transfer coefficients

$$\bar{a}_{ij},\ i=1,\ldots,N,\ j\neq i,$$

where $\bar{a}_{ij}$ is the number of output units from activity A_j required in the production of one output unit by A_i.

Note that in the discrete model, product transfer coefficients are not transfers per unit intensity. However, as the intensity of activity A_i is increased, more frequent product transfers would be required of A_j, implying greater intensity for A_j. Such activity interaction is entirely similar in the case that product transfers are modeled directly proportional to activity intensity (continuous flow model, Chapter 3).

Discrete product transfers are intimately related to the notion of precedence in critical path methods. That is, the work package of A_i producing one unit is preceded by the work package of A_j producing $\bar{a}_{ij}$ units. In order to more fully develop this relationship, it will be instructive to turn to critical path techniques, and to consider extensions of such techniques in the spirit of the dynamic activity analysis.

2.2. Activity Analysis Extensions of Critical Path Techniques

2.2.1. Review of CPM

PERT and CPM are network planning techniques for project management. These techniques are usually applied to one-time efforts, in which similar work may have been done previously, but it is not being repeated on a production basis.

Critical path methods involve both a graphical portrayal of the interrelationships among the activities of a project, and an arithmetic procedure which identifies the contribution of each activity to the overall schedule, as briefly discussed below.

In a CPM project network, nodes denote activities which require time, manpower, and facilities to complete, as illustrated in Figure 2.1.* The arrows indicate work flow dependencies, eg, activity A_2 must be completed before activity A_4 is performed. To maintain correct network logic, cycles are prohibited. An activity is thus any portion of the work that may not begin until other portions are completed.

Each CPM activity A_i has a time assignment t_i which is an estimate of the duration required to complete that portion of project work represented by the activity. The basic scheduling computations are a *forward pass* and a *backward pass* through the network. Typically, start times for the initial activities (in the figure, A_1 and A_2) are given data, so that the forward pass is performed first. Based on the given start times and activity durations, the forward pass computes the *earliest start* and *earliest finish* times for each node (activity). Similarly, when finishing times for final nodes (in the figure, A_7 and A_8) are specified or

* In most presentations of CPM, the roles of nodes and arcs are reversed from the roles used here. Either format can be used in practice. See reference [5].

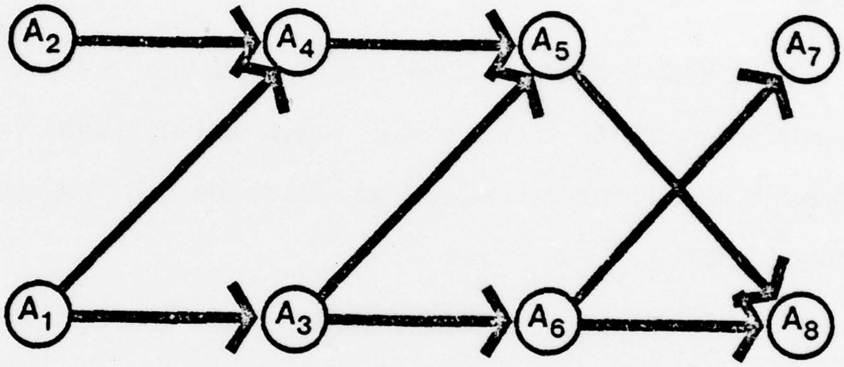


Figure 2.1
Example of CPM Network

computed by the forward pass, the backward pass can compute the *latest start* and *latest finish* times for each node. The times are latest in the sense that if any were later, project finishing times would not be met.

Comparison of backward and forward pass results will indicate that certain paths through the network will be *critical.* That is, start and finish times for activities on these paths any later than the earliest times will cause later project finishing times. Other paths will have *slack,* whereby nodes on such paths could have larger time assignments without impacting these finishing times. Alternatively, the starting times for such nodes (activities) could be delayed.

In general, the critical path P between an initial node and a final node is the longest path through the network connecting those nodes, where the length of the path $T(P)$ is the sum of the time assignments of nodes on the path P. The slack of an alternate path P' between the same two nodes is given by $T(P)-T(P')$. The scheduling slack s_i of a node A_i is the minimum slack value of all slack paths containing that node, and is given by the difference between the late and early start (or finish) times for A_i.

Associated with each activity are requirements for manpower and other resources. A determination of start and finish times for all the activities allows the computation of resource demands by the project for a given schedule. The variation in resource loading resulting from shifting slack activities between early and late start can be studied. See references [5,6,12].

2.2.2. Activity Intensities

The fixed time and resource assignments to each activity in critical path techniques imply fixed application rates of resources. Ordinarily, in a construction project, one has the flexibility to vary such rates to accomplish leveling of resource utilization, balancing of interdependent processes, etc. As a first extension of critical path methods, we shall make the resource assignments to each activity variable, according to the following restricted dynamic activity analysis.

We consider a critical path network of N activities $A_1, \ldots, A_N$ modeling a construction project in which NK exogenous resources are utilized. The rate of application of resources to each activity A_i can be modeled linearly in terms of a strictly positive intensity variable z_i taken with technical coefficients defined as follows:

(a) c_i = fraction of activity A_i completed per unit intensity, per unit time, $i=1, \ldots, N$.

(b) a_{ik} = amount of exogenous resource k applied to A_i per unit intensity, per unit time, $i=1, \ldots, N$, $k=1, \ldots, NK$.

The intensity assignment z_i to activity A_i is maintained at a constant value between start and finish times for A_i, but the particular level of intensity is a decision variable. In this way, the time assignment to each activity A_i becomes a variable, given by

$$t_i = \frac{1}{z_i c_i}, \quad i=1, \ldots, N. \tag{2.1}$$

Typically, for each activity A_i, there is some maximum practical rate at which that activity may operate, implying an *intensity bound* $\bar{z}_i$. These bounds reflect the conditions peculiar to each activity such as available workspace.

Up to this point, this analysis resembles CPM time-cost trade-off procedures (see reference [5]), in which activity durations may vary between "crash" and "normal" durations, and there is a linear relation between activity duration and costs of resources used. However, in our formulation, the relationship between the level of resources demanded by each activity and activity duration is made explicit, so that between the start and finish times for an activity A_i, the rate of application of resource k to activity A_i is given by

$$z_i a_{ik} = \frac{a_{ik}}{t_i c_i}, \quad k=1, \ldots, NK, \quad i=1, \ldots, N. \tag{2.2}$$

Pictorially, the "resource box" assigned to an activity can be "flattened and lengthened" (while keeping the area constant) by decreasing activity intensity, as shown in Figure 2.2.

By developing technical coefficient data and defining intensities, one can then vary activity intensities to accomplish time substitution of activity loading. This approach contrasts with pure scheduling strategies, which seek to minimize resource peaks by shifting a fixed box for each activity between early and late starting times. (See references [3] and [5] for discussions of scheduling strategies; see sections 2.6 and 2.7 for investigations of smoothing and resource leveling.)

An initial selection of activity intensities

$$z_i = z_i^0, \quad i=1, \ldots, N, \tag{2.3}$$

allows the ordinary CPM scheduling calculations of early (late) start and finish times and activity slack, using the node durations

$$t_i = \frac{1}{z_i^0 c_i}, \quad i=1, \ldots, N. \tag{2.4}$$

In particular, operating all activities at their intensity bounds, ie,

$$z_i^0 = \bar{z}_i, \quad i=1, \ldots, N, \tag{2.5}$$

yields a minimum project time span.

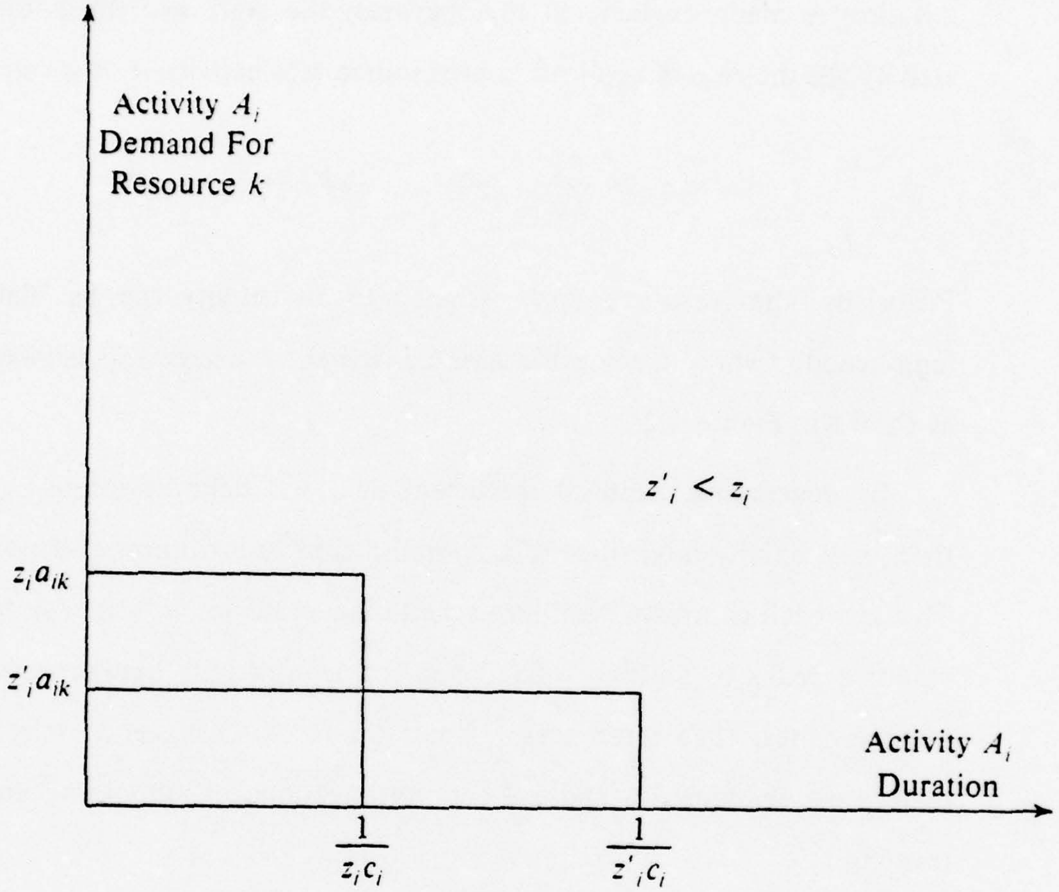


Figure 2.2
Effects of Decreasing Activity Intensity

The intensity z_i^0 of activity A_i is said to be *critical* if the time assignment to A_i,

$$\left(z_i^0 c_i\right)^{-1},$$

is such that activity A_i is on a critical path in the activity network. That is, a lower intensity assignment to activity A_i would extend the project time span.

Any activity A_i may be made critical by lowering its intensity assignment z_i^0 to

$$\left[\frac{1}{z_i^0} + s_i^0 c_i\right]^{-1},$$

where s_i^0 is the slack of activity A_i calculated by the CPM scheduling computations for the time assignments (2.4). Other activities on the same slack path would then also be made critical, as their slack would be eliminated. In general, the slack of a given slack path through the network can be allocated among the activities on the path, whereby the reduced intensity assignments would become critical. Eliminating all activity slack in the network leaves all activities critical.

Reducing activity intensities to criticality tends to smooth project resource loading histories, as the resource demands of slack activities are reduced to lower levels. See the discussion and examples in sections 2.6, Smooth Loading, and 2.7, Resource Leveling. See also reference [11] for further examples.

2.3. Extension to Output Streams

Critical path techniques are oriented to the single construction project scenario. But consider a production system producing a stream of outputs, such as a shipyard producing a sequence of ships. One could develop an activity network for each ship, plan the resource loads, and then attempt to overlay these loads to predict overall yard resource demands.

But desirable elements of the work flow and planning flexibility may be missing from this approach. Consider the simplistic network shown in Figure 2.3 for producing a ship (activities are nodes, arcs show precedence). After activity A finishes work on the first ship, activity B and activity C start work on the first ship. But in view of the sequence of ships to be produced, activity A would then commence work on the second ship. An *extended network* is immediately suggested, whose nodes may be denoted by (i,m), where i denotes the activity, and m denotes the output unit. For three ships produced by the above network, one has the extended network shown in Figure 2.4.

There can now be an intensity assignment z_{im} to activity A_i working on output unit m, which corresponds to a time assignment

$$t_{im} = \frac{1}{z_{im} c_i} \qquad (2.6)$$

to node (i,m) of the extended network.

Technical coefficients can also vary by output unit to admit *learning curve behavior,* whereby either the level of resources required per unit intensity declines or the output per unit intensity per unit time increases. The coefficients are modified as follows:

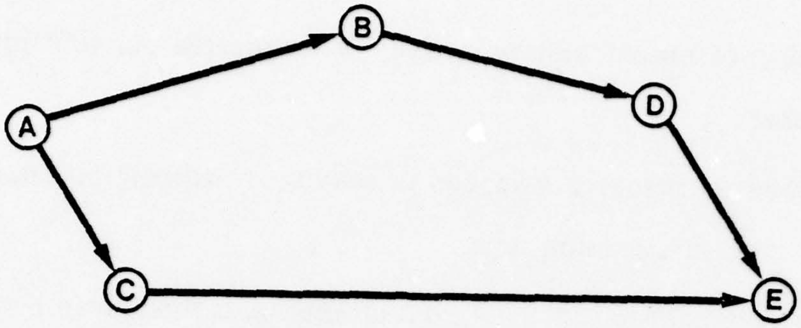


Figure 2.3
Simplistic Shipbuilding Network

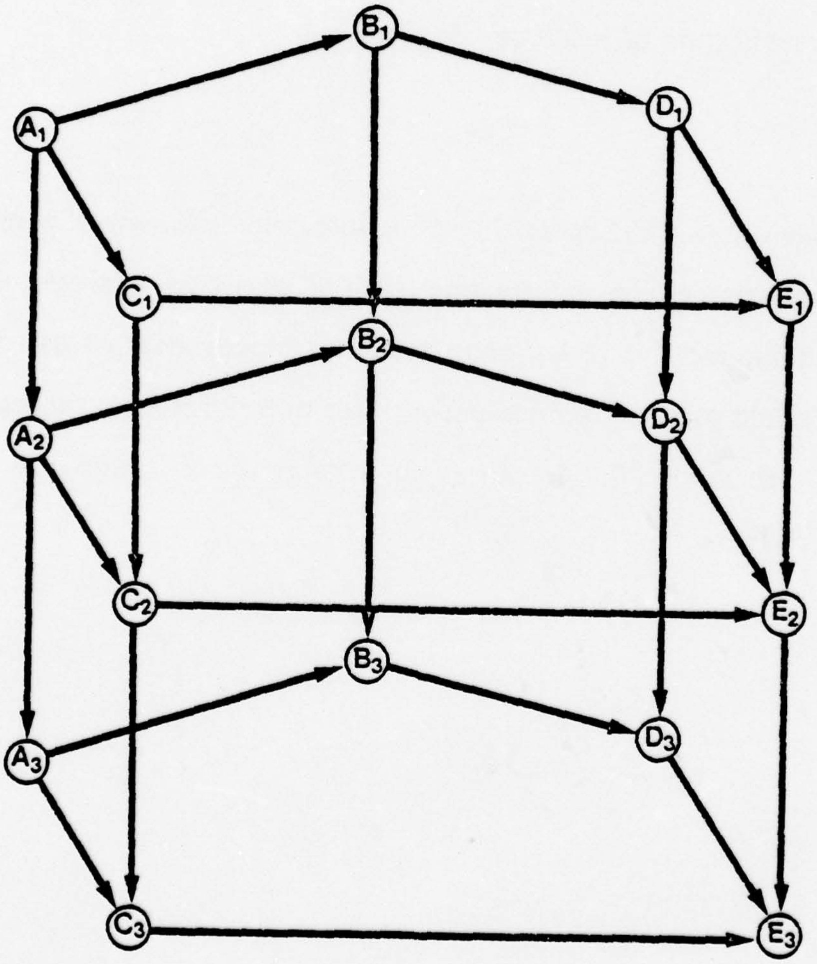


Figure 2.4
Extended Network For Production of Three Ships

(a) c_{im} = fraction of output unit m, activity A_i completed per unit intensity, per unit time;

(b) a_{ikm} = amount of resource k applied to activity A_i working on output unit m per unit intensity, per unit time.

The time assignment to activity A_i, output unit m, is then given by

$$t_{im} = \frac{1}{z_{im} c_{im}} , \qquad (2.7)$$

and the rate of application of resource k is given by

$$z_{im} a_{ikm} .$$

The extended network concept allows the integrated planning of a series of construction projects, effectively treating such a series as a single project. Rather than attempting to overlay or to sequence independently-treated CPM networks, one could use an extended network for overall resource management and allocation. See section 2.7 for an example of resource leveling on a two-ship extended network.

2.4. General Dynamic Model with Discrete Transfers

We now return to the dynamic activity analysis, whereby activities have *intensity time histories*

$$\left\{z_i(t)\right\}$$

defined over a discrete time grid $t=0, 1, 2, \cdots$. Technical coefficients and intensity bounds are also allowed to be variable with time. We shall assume that activity output units are so large that

$$\bar{z}_i(t)c_i(t) < 1 \qquad (2.8)$$

for all activities A_i, $i=1, \ldots, N$, *and* $t=0, 1, 2, \cdots$. (If such is not the case, a finer time grid may be considered.)

We shall also assume the activity network representation of the production system is cycle-free. Product transfer coefficients $\left\{\bar{a}_{ij}\right\}$ are taken to be constant, so that the arcs of the network, while identifying intermediate product transfers, also indicate the precedence of work flow. An arc from activity A_j to activity A_i means that that A_j must complete $(n)\bar{a}_{ij}$ output units before activity A_i can start production of its nth output unit.

By suitable modification, the extended network discussed in section 2.3 can be used to explicitly display each discrete product transfer. For each activity A_i and output unit m, the immediate predecessors of node (i,m) in the extended network are revised to be

$(i,m-1)$, except when $m=1$; and

$(j,\bar{a}_{ij},m)$, except when $\bar{a}_{ij}=0$.

In section 2.3, the extension of a critical path network in order to model output streams is an example of an extended network in which all $\bar{a}_{ij}$ equal zero or one.

Arcs in the activity network are allowed to represent both critical path-style precedences or true product flow. For example, suppose activities A_i and A_j occupy the same workspace, but the installation of A_i product is required before the installation of A_j product may occur. In terms of the model, we can think of activity A_j requiring the workspace *with activity A_i product installed* as intermediate product input. With this interpretation, an activity, which has a critical path-style precedence relationship to m successor activities, simultaneously produces m distinct products according to its intensity. Each of the m arcs emanating from the activity represents the transfer of a distinct product.

For other activities, the conception of product flow may be more immediate, such as the case where activity A_i produces subassemblies to be integrated into the structures produced by activity A_j. It may be the case that the subassemblies produced by A_i supply several activities, so that arcs emanating from A_i show flows of the same product to be shared by successor activities. Calculations for each of these cases of product flow are presented in section 2.5.

Inventories of intermediate products may be subject to storage capacities. In general, there is a storage capacity for each product type. For the presentation of calculations in the next section, we shall initially assume that the arcs of the network represent critical path-style precedences, ie, each arc represents the transfer of a distinct product. Accordingly, we denote

$$cap(i,j),\ i=1,\ .\ .\ .\ ,N,\ j \geqslant i,$$

to be the capacity of stored output of activity A_i to be used by activity A_j, where stored output is the number of units started by A_i, less the number of output units started by A_j times $\bar{a}_{ji}$.* To be feasible, a production plan

$$\left\{ z_i(t) \right\}_{i=1,\ldots,N}^{t=0,\ldots,T-1}$$

must satisfy inventory capacity constraints as well as constraints governing the adequacy of intermediate product transfers. These are discussed in section 2.5.

* In the case that capacity of product storage is unconstrained, the capacity may be set equal to some very large number for programming purposes.

2.5. Dynamic Production Correspondences

Dynamic production correspondences relate time histories of final outputs to time histories of exogenous resource demands. Their abstract structure is discussed in references [7,8,9].

Two kinds of correspondence calculations appropriate for production planning are investigated in this section. The presentation is a generalization of that in reference [11], in which product transfer coefficients were taken to be binary, and product inventories were not explicitly considered.

2.5.1. Forward Correspondence

The forward correspondence addresses the following question: for given time histories of the input of exogenous resources, determine the time histories of output unit completions possible. Let

$$\vec{X}(t) = \left\{X_1(t), \ldots, X_{NK}(t)\right\}, \quad t=0, \ldots, T-1, \tag{2.9}$$

be the given time histories of resources available for production input during $[0,T)$. We consider a *preallocation* of each resource k among the activities, denoted by

$$\overrightarrow{XO}(t) = \left\{\overrightarrow{XO}_1(t), \ldots, \overrightarrow{XO}_{NK}(t)\right\}, \quad t=0, \ldots, T-1, \tag{2.10}$$

where

$$\overrightarrow{XO}_k(t) = \left\{XO_{1,k}(t), \ldots, XO_{N,k}(t)\right\}, \quad k=1, \ldots, NK,$$

$$t=0, \ldots, T-1, \tag{2.11}$$

and

$$\sum_{i=1}^{N} XO_{i,k}(t) = X_k(t)\ , \ k=1, \ldots, NK,\ t=0, \ldots, T-1. \qquad (2.12)$$

The full possibilities of production result from the union of the sets of network output histories obtainable from all preallocations $\overline{XO}(t)$ of $\overline{X}(t)$.

An N-by-N incidence matrix

$$\left\{D(i,j)\right\}$$

will be used to describe the network, with $D(i,j)=1$ if there is an arc in the activity network from activity A_i to activity A_j (ie, $\bar{a}_{ji} > 0$), and $D(i,j)=0$ if not. As the network is cycle-free, we shall assume the activity ordering is such that A_i may serve only activities A_j, $j>i$, $i=1, \ldots, N$.

The following notation will be convenient. Let

$M_t(i)$ = number of output units *completed* by activity A_i by time t;

and $N_t(i)$ = number of output units *started* by activity A_i by time t.

With this notation, when activities A_i and A_j are connected by an arc (ie, $D(i,j)=1$), the time history of intermediate product inventory of A_i to be used by A_j is given by

$$inv_t(i,j) = M_t(i) - \bar{a}_{ji} N_t(j). \qquad (2.13)$$

The constraints on intensity functions for the forward correspondence are as follows:

F(1). $z_i(t) \geqslant 0,\ \ i=1, \ldots, N,\ t=0, \ldots, T-1.$

F(2). $z_i(t) \leqslant \bar{z}_i(t),\ \ i=1, \ldots, N,\ t=0, \ldots, T-1.$

F(3). $a_{ik}(t)z_i(t) \leqslant XO_{ik}(t)$, $i=1,\ldots,N$, $t=0,\ldots,T-1$.

F(4). $\bar{a}_{ij}c_i(t+1)z_i(t) \leqslant M_t(j) - \sum_{\tau=0}^{t-1}\bar{a}_{ij}c_j(\tau+1)z_i(\tau)$,

$i=1,\ldots,N$, *all j such that* $D(j,i)=1$, $t=1,\ldots,T-1$, *and*

$\bar{a}_{ij}c_i(1)z_i(0) \leqslant 0$, $i=1,\ldots,N$, *all j such that* $D(j,i)=1$.

F(5). $c_i(t+1)z_i(t) \leqslant cap(i,l) + \bar{a}_{li}N_{t+1}(l) - \sum_{\tau=0}^{t-1}c_i(\tau+1)z_i(\tau)$,

$i=1,\ldots,N$, *all l such that* $D(i,l)=1$, $t=1,\ldots,T-1$, *and*

$c_i(1)z_i(0) \leqslant cap(i,l)$, $i=1,\ldots,N$, *all l such that* $D(i,l)=1$,

where

$$M_t(i) = Max\left\{n\ integer \mid n \leqslant \sum_{\tau=1}^{t} c_i(\tau)z_i(\tau-1)\right\},$$

$t=1,\ldots,T$, (2.14)

$$M_0(i) = 0, \tag{2.15}$$

and

$$N_t(i) = Min\left\{n\ integer \mid n \geqslant \sum_{\tau=1}^{t} c_i(\tau)z_i(\tau-1)\right\},$$

$t=1,\ldots,T$, (2.16)

$$N_0(i) = 0. \tag{2.17}$$

The constraints F(1) and F(2) allow non-negative values for intensities less than or equal to the upper bounds. Constraints F(3) limit intensities to levels allowed by preallocation. Constraints F(4) limit intensities to levels allowed by available intermediate products, while constraints F(5) insure that intermediate product inventories do not exceed capacity.

A production plan satisfying constraints F(1) through F(5) is said to be *forward feasible*. Clearly, the output histories one can obtain are not unique. As an initial approach, we shall consider a *greedy policy*, ie, when an activity is called upon to operate, it runs at maximal intensity. Such a policy may be calculated forward in time, performing the four steps of the algorithm shown below at each instant of time. It will be convenient to keep track of cumulative production using the variables

$$cum_t(i), \ i=1, \ldots, N, \ t=1,2, \ldots, T,$$

where

$$cum_t(i) = \sum_{\tau=0}^{t} z_i(\tau) c_i(\tau+1). \tag{2.18}$$

The *greedy algorithm* for the forward correspondence is then as follows:

Step F0. *Initialize* $M_0(i) = N_0(i) = cum_0(i) = 0, \ i=1, \ldots, N,$

and set $t=0$.

Step F1. *For* $j=1, \ldots, N,$ *set* $z_j(t)=$

$$Min\left\{\bar{\bar{z}}_j(t),\ \underset{k=1,\ldots,NK}{Min}\left[\frac{XO_{jk}(t)}{a_{jk}(t)}\right],\ \underset{\substack{all\ i\\ such\ that\\ D(i,j)=1}}{Min}\left[\frac{M_t(i)-\bar{a}_{ji}cum_t(j)}{\bar{a}_{ji}c_j(t+1)}\right]\right\}.$$

Step F2. *Set* $z_N(t) = \bar{\bar{z}}_N(t)$.

Step F3. *For* $i=N-1, N-2, \ldots, 1,$ *set* $z_i(t)=$

$$Min\left\{\bar{\bar{z}}_i(t),\ \underset{\substack{all\ i\\ such\ that\\ D(i,j)=1}}{Min}\left[\frac{cap(i,j)+\bar{a}_{ji}\left[\beta(j)+N_t(j)\right]-cum_t(i)}{c_i(t+1)}\right]\right\},$$

where

$$\beta(j) = \begin{cases} 1 & \text{if } z_j(t)c_j(t+1)+cum_t(j) > N_t(j) \\ 0 & \text{if } not \end{cases}, \qquad (2.19)$$

an indiciator variable, shows whether activity A_j initiated production of a new output unit during the current time period.

Step F4. *Set* $N_{t+1}(j) = N_t(j) + \beta(j)$, *all j,*

where $\beta(j)$ is defined by (2.19). Set

$$M_{t+1}(j) = M_t(j) + \alpha(j),\ all\ j,$$

where

$$\alpha(j) = \begin{cases} 1 & \text{if } z_j(t)c_j(t+1) + cum_t(j) > M_t(j)+1 \\ 0 & \text{if } not \end{cases}, \qquad (2.20)$$

is an indicator variable, showing whether activity A_j completed production of an output unit during the current time period. Set

$$cum_{t+1}(i) = cum_t(i)+z_i(t)c_i(t+1),\ i=1,\ldots,N.$$

Increment $t \rightarrow t+1$. Stop if $t=T$; otherwise, go to Step 1.

Steps F1 and F2 insure the intensities do not exceed levels allowed by intensity bounds, available exogenous inputs and available intermediate products. Step F3 insures that intermediate product inventories do not exceed capacity. Step F4 updates inventories and increments time.

From the calculated intensity histories, one can observe the earliest starting and finishing times of each output unit by each activity. In fact, the array

$$\left\{cum_i(t)\right\}_{t=0,\ldots,T}^{i=1,\ldots,N}$$

indicates the time unit by time unit progress of each activity.

One can also obtain the system use of resources from the intensity histories. During the time period $[t,t+1)$, the actual use of resource k is given by

$$\sum_{i=1}^{N} z_i(t)a_{ik}(t).$$

In the case where activity output supplies both intermediate and final (net) product, or in the case where activity product is shared by two or more follow-on activities, a preallocation must also be made of such output. The constraints on intensity functions and the forward greedy algorithm can easily be modified to handle the above cases.

As an example, suppose activity A_π produces one product which supplies final output as well as activities A_γ and $A_{\gamma+1}$. Let $\delta_\pi(t)$, $t=1,2,\ldots,T$, denote the fraction of the output history of A_π going to final output at time t, where $0 \leqslant \delta_\pi(t) \leqslant 1$. Let $\Delta_{\pi i}(t)$, $t=1,2,\ldots,T$, $i=\gamma$, $\gamma+1$, be non-negative real numbers denoting a preallocation of A_π output at time t among A_γ and

$A_{\gamma+1}$ as intermediate product input. These coefficients satisfy

$$\Delta_{\pi\gamma}(t) + \Delta_{\pi\gamma+1}(t) = 1 - \delta_{\pi}(t), \quad t=1, \ldots, T. \tag{2.21}$$

For simplicity of exposition, we shall assume no activities other than A_{π} supply A_{γ} and $A_{\gamma+1}$. We denote

$$cap(\pi)$$

to be the capacity of stored output of A_{π} to be used by A_{γ} and $A_{\gamma+1}$, where stored output is the number of output units for intermediate uses started by A_{π} less the number of output units started by A_{γ} times $\bar{a}_{\gamma\pi}$, and less the number of output units started by $A_{\gamma+1}$ times $\bar{a}_{\gamma+1,\pi}$.

The forward correspondence constraints F(1), F(2), and F(3) on intensity functions require no modification, but constraints F(4) and F(5) must be altered as follows:

F(4)′. *Same as* F(4) for $i \neq \gamma$, $\gamma+1$. *For* $i=\gamma, \gamma+1$, *we have*

$$\bar{a}_{i\pi} c_i(t+1) z_i(t) \leqslant M_t(\pi,i) - \sum_{\tau=0}^{t-1} \bar{a}_{i\pi} c_i(\tau+1) z_i(\tau),$$

$t=1, \ldots, T-1$, *and*

$$\bar{a}_{i\pi} c_i(1) z_i(0) \leqslant 0.$$

F(5)′. *Same as* F(5) for $i \neq \pi$. *For* $i=\pi$, *we have*

$$\sum_{l=\gamma}^{\gamma+1} \Delta_{\pi l}(t+1) c_{\pi}(t+1) z_{\pi}(t) \leqslant$$

$$cap(\pi) + \sum_{l=\gamma}^{\gamma+1} \bar{a}_{l\pi} N_{t+1}(l) - \sum_{l=\gamma}^{\gamma+1} \sum_{\tau=0}^{t-1} \Delta_{\pi l}(\tau+1) c_{\pi}(\tau+1) z_{\pi}(\tau),$$

$t=1, \ldots, T-1$, *and*

$$\sum_{i=\gamma}^{\gamma+1} \Delta_{\pi i}(1) c_{\pi}(1) z_{\pi}(0) \leqslant cap(\pi),$$

where

$$M_t(\pi,i) = Max\left\{n \ integer \,\middle|\, n \leqslant \sum_{\tau=0}^{t-1} \Delta_{\pi i}(\tau) c_{\pi}(\tau+1) z_{\pi}(\tau)\right\},$$

$t=1, \ldots, T,$ (2.22)

$$M_0(\pi,i) = 0, \tag{2.23}$$

and

$$N_t(i) = Min\left\{n \ integer \,\middle|\, n \geqslant \sum_{\tau=0}^{t-1} c_i(\tau+1) z_i(\tau)\right\},$$

$t=1, \ldots, T,$ (2.24)

$$N_0(i) = 0. \tag{2.25}$$

The steps of the forward greedy algorithm are modified as follows:

Step F0′. *Same as step* F0., *except initialize* $M_0(\pi,i)=0$, $i=\gamma, \gamma+1$, *in place of* $M_0(\pi)$.

Step F1′. *Same as step* F1., *except* for $j=\gamma$, $\gamma+1$, *set* $z_j(t) =$

$$Min\left\{\Xi_j(t), \ \underset{k=1,\ldots,NK}{Min}\left[\frac{XO_{jk}(t)}{a_{jk}(t)}\right], \ \frac{M_t(\pi,j) - \bar{a}_{j\pi} cum_t(j)}{\bar{a}_{j\pi} c_j(t+1)}\right\}.$$

Step F2$'$. *Same as step* F2.

Step F3$'$. *Same as step* F3., *except* for $i=\pi$, *we have* $z_\pi(t) =$

Minimum of $\bar{z}_\pi(t)$ and

$$\frac{cap(\pi) + \sum_{j=\gamma}^{\gamma+1} \bar{a}_{j\pi}\left[\beta(j)+N_t(j)\right] - \sum_{j=\gamma}^{\gamma+1}\sum_{\tau=0}^{t-1} \Delta_{\pi j}(\tau+1)c_\pi(\tau+1)z_\pi(\tau)}{\sum_{j=\gamma}^{\gamma+1} \Delta_{\pi j}(t+1)c_\pi(t+1)},$$

where $\beta(j)$ is defined by (2.19).

Step F4$'$. *Same as step* F4., *except in place of* $M_{t+1}(\pi)$, *we*

set

$$M_{t+1}(\pi,l) = M_t(\pi,l) + \alpha(\pi,l),\ l=\gamma,\gamma+1,$$

where

$$\alpha(\pi,l) = \begin{cases} 1 \text{ if } \sum_{\tau=0}^{t} \Delta_{\pi l}(\tau+1)c_\pi(\tau+1)z_\pi(\tau) > M_t(\pi,i)+1 \\ 0 \text{ if } \textit{not.} \end{cases} \tag{2.26}$$

Production systems in which an activity produces several products, each of which serves several follow-on activities, are handled in an analagous manner, whereby each product must be preallocated among uses. For simplicity of exposition, in algebraic presentations in the remainder of Chapter 2, it is assumed that each arc in the production network represents the transfer of a distinct product (eg, critical path-style precedences).

The greedy policy can also be calculated starting at a time point with production in progress. This is done by appropriately initializing the arrays of cumulative activity production,

$$\left\{cum_t(i)\right\},\ \left\{M_t(i)\right\},\ and \left\{N_t(i)\right\},$$

in step F0. of the algorithm. In this way production replanning required because of work interruptions or design changes can be accomplished.

As the forward greedy policy assigns activity intensities at the upper bounds allowed by constraints **F**(1) through **F**(5) at each instant of time, the throughput of production is maximal for the preallocations made. The full spectrum of output histories obtainable from the vector $\overline{X}(t)$ of histories of exogenous inputs is only realized when, in addition to the alternatives for preallocating exogenous input histories and transfers of activity outputs as intermediate products and net outputs, one considers also the operation of activities with intensities less than·the maximal values calculated by the greedy policy. In particular, the preallocations, in conjunction with the intensity bounds, will ordinarily be imperfectly balanced so that some activity intensities may be operated at less than the maximal intensities during certain periods and still yield the net output streams calculated by the greedy policy. Such shifts in the intensity functions and the related input histories represent *time substitutions* of the input histories applied. Time substitutions are studied in sections 2.6, Smooth Loading, and 2.7, Resource Leveling.

In the special case that product inventories are not capacity-constrained, and intensity bounds, technical coefficients, and resource preallocations are constant in time, the forward greedy policy is equivalent to a CPM forward pass through the extended network, with time assignments

$$t_{im} = \frac{1}{c_i}\left[Min\left\{\Xi_i,\ \underset{k=1,\ldots\ldots,NK}{Min}\left(\frac{XO_{ik}}{a_{ik}}\right)\right\}\right]^{-1} \qquad (2.27)$$

to each node (i,m) of the extended network. This result also holds when technical coefficients and intensity bounds vary by output unit, as discussed in section 2.3.

2.5.2. Backward Correspondence

The backward correspondence addresses the following question: determine the activity intensities so that given dated sequences of net output units may be obtained. We shall assume one or more activities produce final outputs. Retaining the property that any activity A_i can supply only activities A_j for $j>i$, the ordering of the activities may be made so that A_i, $i=N,N-1, \ldots, N-p+1$ $(p \geqslant 1)$ yield final (net) outputs.

In the case two or more such activities p.roduce the same final output, we shall assume a preallocation of net. output requirements is made among these activities, so that one may express output requirements as time histories

$$\left\{ \hat{U}_i(t) \right\}$$

of *cumulative* output that must be completed by each final activity A_i, $i=N-p+1, \ldots, N$. For example, suppose activity A_N must complete two units by time 25, one more unit by time 30, and a fourth unit by time 45. Then the time history $\left\{ \hat{U}_N(t) \right\}$ is constructed as follows:

$$\hat{U}_N(t) = \begin{cases} 4 \text{ if } t \geqslant 45 \\ 3 \text{ if } 30 \leqslant t < 45 \\ 2 \text{ if } 25 \leqslant t < 30 \\ 0 \text{ if } t < 25 \, . \end{cases} \tag{2.28}$$

The significance of $\left\{\hat{U}_N(t)\right\}$ is clear; if the cumulative production of activity A_N at a given time t does not equal or exceed $\hat{U}_N(t)$, the production plan would be infeasible.

As a first approach it will also be assumed that none of the activities produce the same intermediate product, ie, there are no alternative processes in the production network. This simplifies the calculations, because then one need not attempt to apportion among alternative activities production of product needed by follow-on activities.

Calculations for the backward correspondence are made backwards in time from some time horizon T, by which time all output must be completed. We again assume a preallocation of the services of fixed capital equipment and other resources, denoted by

$$\left\{\overline{XO}_k(t)\right\},\ k=1,\ldots,NK,\ t=T-1,T-2,T-3,\cdots.$$

The constraints on intensity functions for the backward correspondence are then as follows:

B(1). $z_i(t) \geqslant 0,\ \ t=T-1,T-2,\ldots,\ i=1,\ldots,N.$

B(2). $z_i(t) \leqslant \bar{z}_i(t),\ \ t=T-1,T-2,\ldots,\ i=1,\ldots,N.$

B(3). $a_{ik}(t)z_i(t) \leqslant XO_{ik}(t),\ \ t=T-1,T-2,\ldots,\ i=1,\ldots,N,$

$k=1,\ldots,NK.$

B(4). $c_i(t+1)z_i(t) \leqslant \hat{U}_i(T) - \hat{U}_i(t) - \sum_{\tau=t+1}^{T-1} c_i(\tau+1)z_i(\tau),$

$t=T-2, T-3, \ldots, \; i=N-p+1, \ldots, N$, *and*

$$c_i(T)z_i(T-1) \leqslant \hat{U}_i(T) - \hat{U}_i(T-1), \quad i=N-p+1, \ldots, N.$$

B(5). $c_i(t+1)z_i(t) \leqslant \bar{a}_{ji} M^B_{t+1}(j) - \sum_{\tau=t+1}^{T-1} c_i(\tau+1)z_i(\tau),$

$i=1, \ldots, N$, *all* j *such that* $D(i,j)=1, \; t=T-2, T-3, \cdots.$

B(6). $\bar{a}_{il} c_i(t+1)z_i(t) \leqslant cap(i,l) + N_t^B(l) - \sum_{\tau=t+1}^{T-1} \bar{a}_{il} c_i(\tau+1)z_i(\tau),$

$i=2, \ldots, N$, *all* l *such that* $D(i,l)=1, \; t=T-2, T-3, \ldots,$

where

$$M_t^B(i) = Min\left\{ n \text{ integer} \mid n \geqslant \sum_{\tau=t}^{T-1} c_i(\tau+1)z_i(\tau) \right\},$$

$$t=T-1, T-2, T-3, \ldots, \tag{2.29}$$

is the number of output units started by A_i *after* time t, and

$$N_t^B(i) = Max\left\{ n \text{ integer} \mid n \leqslant \sum_{\tau=t}^{T-1} c_i(\tau+1)z_i(\tau) \right\},$$

$$t=T-1, T-2, T-3, \ldots, \tag{2.30}$$

is the number of output units completed by A_i *after* time t.

The constraints **B**(1) and **B**(2) limit activity intensities to non-negative values less than or equal to the intensity bounds. The constraints **B**(3) express the exogenous input limitations due to the preallocations of resources. The constraints **B**(4) require in cumulative terms that the output histories for the final products of the network do not exceed the histories

$$\left\{ \hat{U}_i(t) \right\}, \quad i = N-p+1, \ldots, N,$$

sought for these products. The intensity functions permitted by **B**(4) need not satisfy equality for all periods, since storage of final product is permitted. By extending (backward) the number of time units used, cumulative requirements can be met by storing final products produced earlier. Similarly, constraints **B**(5) require in cumulative terms that output histories of activities producing intermediate products do not exceed requirements demanded by follow-on activities. Again equality signs need not apply for all time periods, due to the possibility of product storage. Constraints **B**(6) insure that intermediate product inventories are not driven beyond capacity.

A production plan satisfying constraints **B**(1) through **B**(6) is said to be *backward feasible*. Clearly, the input histories are not unique. We shall again consider a *greedy policy,* ie, when an activity is called upon to operate, it runs at maximal intensity. Thus the latest starting times for activities which allow the realization of the given output schedule will be calculated. In this way the shortest time spans to produce the given output amounts are obtained. It will be convenient to keep track of cumulative activity production backwards through time using the variables

$$cum_t(i), \quad i = 1, \ldots, N, \quad t = T-2, T-3, T-4, \ldots,$$

where

$$cum_t(i) = \sum_{\tau = t+1}^{T-1} c_i(\tau+1) z_i(\tau). \tag{2.31}$$

The greedy algorithm for the backward correspondence is then as follows:

Step B0. *Initialize* $M_t^B(i) = N_t^B(i) = cum_t(i) = 0,\ i=1, \ldots, N,$

and set $t=T-1$.

Step B1. *For* $i=N-p+1, \ldots, N$, *set* $\bar{z}_i(t) =$

$$Min\left\{\bar{z}_i(t),\ \frac{\hat{U}_i(T)-\hat{U}_i(t)-cum_t(i)}{c_i(t+1)},\ \underset{\substack{all\ j\\ such\ that\\ D(i,j)=1}}{Min}\left[\frac{\bar{a}_{ji}M_t^B(j)-cum_t(i)}{c_i(t+1)}\right]\right\}.$$

For $i=1, \ldots, N-p$, *set* $\bar{z}_i(t) =$

$$Min\left\{\bar{z}_i(t),\ \underset{\substack{all\ j\\ such\ that\\ D(i,j)=1}}{Min}\left[\frac{\bar{a}_{ji}M_t^B(j)-cum_t(i)}{c_i(t+1)}\right]\right\}.$$

Step B2. *Set* $z_1(t) = \bar{z}_1(t)$.

Step B3. *For* $j=2, \ldots, N$, *set* $z_j(t) =$

$$Min\left\{\bar{z}_j(t),\ \underset{\substack{all\ j\\ such\ that\\ D(i,j)=1}}{Min}\left[\frac{cap(i,j)+N_t^B(i)+\alpha(i)-\bar{a}_{ji}cum_t(j)}{\bar{a}_{ji}c_j(t+1)}\right]\right\},$$

where

$$\alpha(i) = \begin{cases} 1 & \text{if } z_i(t)c_i(t+1) + cum_t(i) > N_t^B(i) \\ 0 & \text{if } not \end{cases} \qquad (2.32)$$

an indicator variable, shows whether activity A_i completed an additional output unit after time $t-1$ over that completed after time t.

Step B4. *Set* $N_{t-1}^B(i) = N_t^B(i) + \alpha(i)$, *all* i, where $\alpha(i)$ is defined by (2.32). Set

$$M_{t-1}^{B}(i) = M_t^{\beta}(i) + \beta(i), \ all\ i,$$

where

$$\beta(i) = \begin{cases} 1 & \text{if } z_i(t)c_i(t)+cum_t(i) > M_t^{\beta}(j)+1 \\ 0 & \text{if } not \end{cases}, \tag{2.33}$$

an indicator variable, shows whether activity A_i started an additional output unit after time $t-1$ over that started after time t. Set

$$cum_{t-1}(i) = cum_t(i)+z_i(t)c_i(t+1), \ i=1, \ldots, N.$$

Increment $t \rightarrow t-1$. Stop when

$$\left\{cum_t(i)\right\}, \ i=N-1+p, \ldots, N,$$

are all greater than or equal to the desired number of output units; otherwise, go to Step 1.

Note the strong similarity of the backwards greedy policy calculation to the forward greedy calculation, the former essentially a "backwards through time" version of the latter. The actual cumulative output completion schedules for the activities are given by

$$\left\{\hat{V}_i(t)\right\}, \ t=T, T-1, T-2, \ldots, \ i=1, \ldots, N,$$

where

$$\hat{V}_i(t) = \sum_{\tau=\mu}^{t-1} c_i(\tau+1)z_i(\tau), \ t=T, T-1, \ldots, \ i=1, \ldots, N, \tag{2.34}$$

and μ is the earliest time period at which the backward greedy algorithm makes calculations. The system use of resources is given by

$$\sum_{i=1}^{N} z_i(t)a_{ik}(t)$$

for each resource $k=1, \ldots, NK$, during each time period $[t,t+1)$.

From inspection of the output completion histories

$$\left\{\hat{V}_i(t)\right\},\ t=T,T-1,T-2,\ldots,\ i=1,\ldots,N,$$

the backward greedy policy yields the late start and late finish times for each output unit produced by each activity. This is entirely analogous to the early start/finish data generated in the forward correspondence case.

To extend the backward correspondence results further, suppose two activities producing intermediate product produce the same output, say A_α and $A_{\alpha+1}$, for $1 \leqslant \alpha < N-p$. Then the constraints **B**(5) are modified for $i=\alpha$ and $i=\alpha+1$ to

$$\mathbf{B}(5)'.\ \sum_{i=\alpha}^{\alpha+1} c_i(t+1)z_i(t) \leqslant \bar{a}_{j,\alpha+1}M^B_{t+1}(j) - \sum_{i=\alpha}^{\alpha+1}\sum_{\tau=t+1}^{T-1} c_i(\tau+1)z_i(\tau),$$

all j such that $D(\alpha,j)=1$, $t=T-2,T-3,\cdots$.

Note that coefficients $\bar{a}_{j,\alpha}$ and $\bar{a}_{j,\alpha+1}$ must be identical. In order to accomodate such substitutable activities, some kind of policy rule is needed. Ordinarily, duplicating processes can be ordered by their efficiency, say $A_{\alpha+1}$ is more efficient or more preferred for whatever reason. Then one may replace **B**(5)$'$ by

$$\mathbf{B}(5)''.\ c_{\alpha+1}(t+1)z_{\alpha+1}(t) \leqslant \bar{a}_{j,\alpha+1}M^{B}_{t+1}(j) - \sum_{\tau=t+1}^{T-1} c_{\alpha+1}(\tau+1)z_{\alpha+1}(\tau),$$

all j such that $D(\alpha,j)=1,\ t=T-2,T-3,\ldots,$

and

$$\sum_{\tau=t}^{T-1} c_{\alpha}(\tau+1)z_{\alpha}(\tau) \leqslant \bar{a}_{j,\alpha+1}M^{B}_{t+1}(j) - \sum_{\tau=t}^{T-1} c_{\alpha+1}(\tau+1)z_{\alpha+1}(\tau),$$

all j such that $D(\alpha,j)=1,\ t=T-2,T-3,\cdots.$

With the addition of such a policy rule, a backward greedy solution is then made determinable for a system with alternate processes.

The full spectrum of exogenous input histories which can achieve the output schedules

$$\left\{\hat{U}_i(t)\right\},\ i=N-p+1,\ldots,N,$$

is only realized when, in addition to the alternatives for preallocation of final output and intermediate product among alternative activities, and of exogenous resources among the activities, one considers also the operation of activities with intensities less than the maximal values calculated by the greedy policy. In particular, reduced intensity assignments allowing more favorable activity and resource loading without violating project due dates are investigated in the following sections.

In the special case that product inventories are not capacity-constrained, and intensity bounds, technical coefficients, and resource preallocations are constant in time, the backward greedy policy is equivalent to a CPM backward pass through the extended network, with time assignments

$$t_{im} = \frac{1}{c_i}\left[Min\left\{\Xi_i, \underset{k=1,\ldots,NK}{Min}\left(\frac{XO_{ik}}{a_{ik}}\right)\right\}\right]^{-1} \qquad (2.35)$$

to each node (i,m) of the extended network. Late finish times for nodes (i,m), $i=N-p+1, \ldots, N$, and the various values of m, are specified by the due date histories

$$\left\{\hat{U}_i(t)\right\},\ i=N-p+1, \ldots, N.$$

That is, a cumulative output requirement of m units by time T from activity N means node (N,m) has a late finish time T. For each such finishing node, there are critical paths through the extended network to the finishing node which can be determined by usual CPM procedures.

The above result also applies in the case technical coefficients and intensity bounds vary by output unit (see section 2.3), whereupon the time assignment to node (i,m) is given by

$$t_{im} = \frac{1}{c_{im}}\left[Min\left\{\Xi_{im}, \underset{k=1,\ldots,NK}{Min}\left(\frac{XO_{ik}}{a_{ikm}}\right)\right\}\right]^{-1}. \qquad (2.36)$$

2.5.3. Inventories and The Extended Network

We have seen that when intensity limitations and technical coefficients are constant in time, and there are no intermediate product inventory capacity constraints, the forward and backward correspondences may be calculated using CPM techniques on the extended network.

It is also possible to use such path methods when there exist capacities for such inventories by making some modifications to the extended network. For example, suppose activity A_i supplies activity A_j, and the inventory capacity is n

units. This means activity A_i cannot have completed any greater number of output units than n more than activity A_j has used as input, ie, node (j,k) must start no later than node $(i,k\cdot\bar{a}_{ji}+n)$ starts, $k=1,2,\ldots$. This implies all predecessor nodes in the extended network of (j,k) also precede node $(i,k\cdot\bar{a}_{ji}+n)$. If the extended network is modified to show these extra precedences, the network logic will then prevent inventories from exceeding capacities.

In summary, for the dynamic production system model with discrete product transfers, the forward and backward greedy policies constitute a generalization of critical path procedures, in which activities have variable intensities, output streams are allowed, and product inventories with capacities are treated. However, it shares the severe limitation with CPM that a preallocation of resources is made, determining maximal activity output rates.

2.6. Smooth Loading

For either forward or backward correspondences, an initial greedy calculation will yield a production plan with the shortest time span to produce a given number of output units, for the preallocation of resources among the activities that is specified. However, such schedules can have undesirable, jagged activity loading characteristics, and one would want to reduce loading levels for smoother behavior without extending the project span.

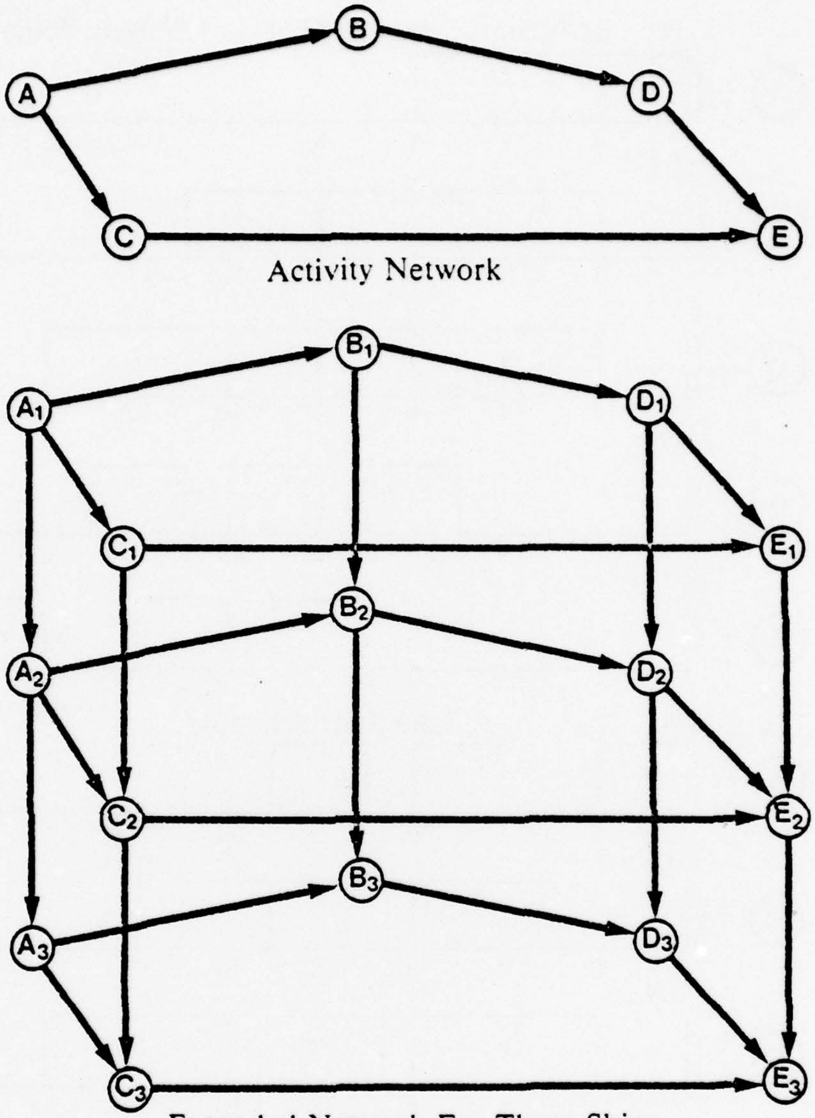
Hereafter we shall assume resource preallocations and intensity bounds are constant in time, and technical coefficients are also constant in time or perhaps vary by output unit, indicating learning curve or other behavior. These restrictions facilitate the study of the smoothing problem in terms of the extended network. Before presenting an algorithm for smoothing, we shall consider a simple example.

Returning to the simplistic shipbuilding network presented in section 2.2, we consider the extended network for building three ships with some specific technical data and intensity bounds, as shown in Figure 2.5.

The activity intensity histories are sketched for the early start and late start schedules in Figure 2.6. Starred "boxes" indicate output units during which activity intensities are critical. Note that an activity may be critical during production of certain output units and non-critical at other times. In this case, the shortest time span is 13 time units.

We see that, in both schedules, "gaps" exist in the intensity histories, where loading ceases for an interval, and then recommences. It is apparent that serial connection of activities with different output rates causes jagged loading.

In the spirit of critical path techniques, one could consider shifting production activity between early and late start schedules to achieve less jagged loading. Indeed, for this simple example, it is evident that a combination of early



Activity Network

Extended Network For Three Ships

Output Rates: $c_A = c_B = c_C = c_D = c_E = 1$

Intensity Bounds: $\bar{z}_A = \frac{1}{2}, \bar{z}_B = \frac{1}{3}, \bar{z}_C = \frac{1}{4}, \bar{z}_D = \frac{1}{2}, \bar{z}_E = \frac{1}{2}$

Figure 2.5
Production Data For Simplistic Shipbuilding Network

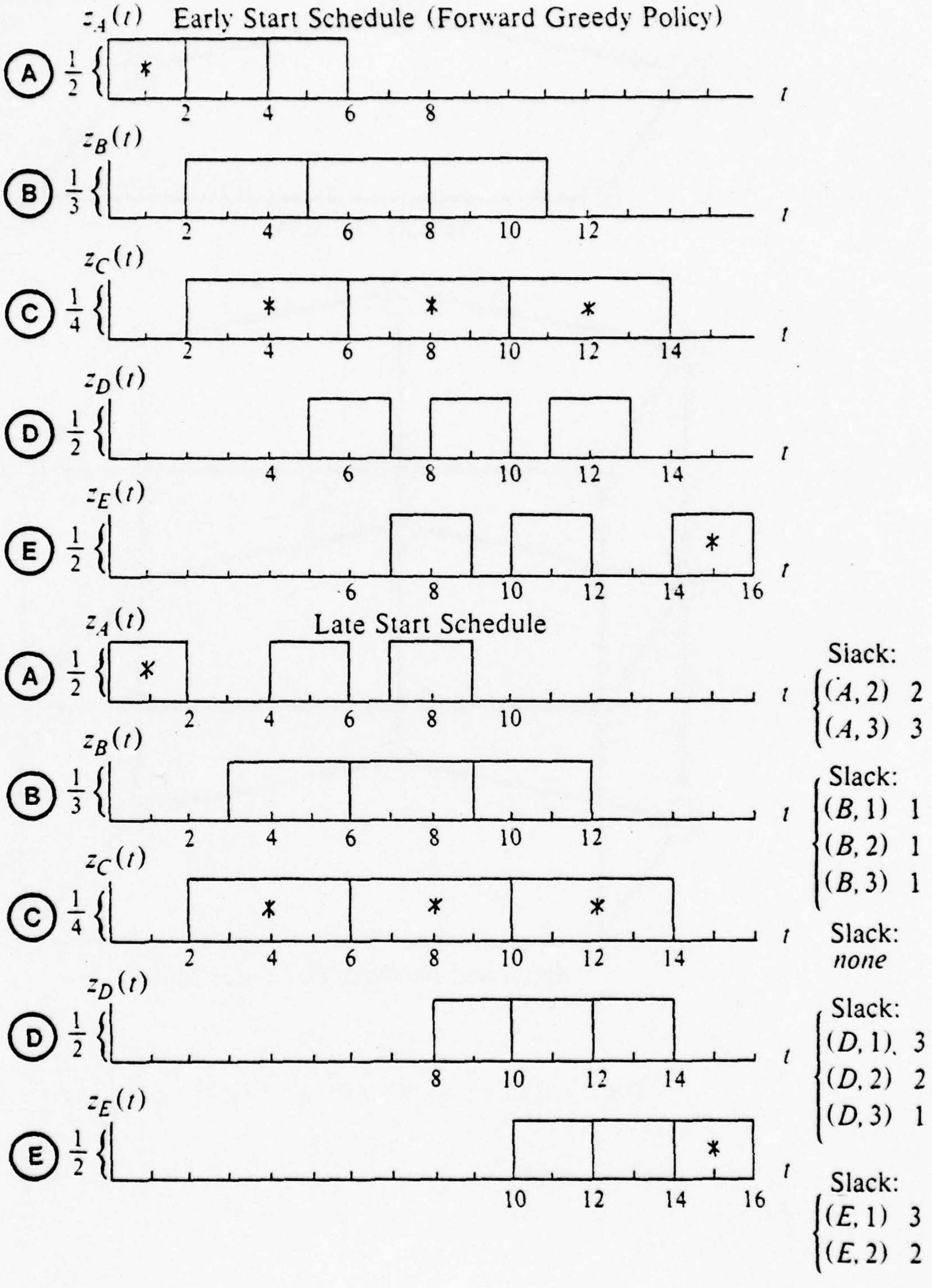


Figure 2.6
Intensity Histories For Production of Three Ships

and late start schedules can produce level activity loading (eg, activities A and B early start, activities C,D, and E late start). But such a schedule does not exploit the fact that activities A, B, D, and E are noncritical at certain times, and therefore resource loading levels of these activities could be reduced.

The duration per output unit of activity D could be increased from two to three time units (ie, $z_{D,1}$, $z_{D,2}$, and $z_{D,3}$ could be reduced from $\frac{1}{2}$ to $\frac{1}{3}$) without violating the critical path schedule limitations. A lower level of resources could then be committed to activity D without slipping the schedule. However, such a change would eliminate slack for activity B, and thus no reduction in intensity of activity B could then avoid extending the project span. Alternatively, the intensity of activity B could first be reduced from $\frac{1}{3}$ to $\frac{3}{10}$, but then the intensity of activity D working on the third output unit ($z_{D,3} = \frac{1}{2}$) would become critical.

It is clear that there is no unique procedure for smoothing unless there is a priority or weighting scheme for the activities, ie, some clear mathematical objective. We shall consider the natural goal of smoothing to be to *minimize activity intensity peaks,* where activities are weighted according to the following formulation, in which output coefficients have been generalized to admit learning effects.

Let p_i be the *smoothing weight* for activity A_i, $i=1, \ldots, N$. Each p_i can be thought of as a capacity cost for activity A_i. Let

$$\left\{Z_i\right\}, \; i=1, \ldots, N,$$

denote the peak intensities encountered by the activities. We consider a network of N activities each producing NM output units. Using the notation of section 2.3, in which z_{im} denotes the intensity assignment to activity A_i working on output unit m, we wish to

$$\textit{Minimize} \quad \sum_{i=1}^{N} p_i Z_i$$

subject to

$$(i) \quad z_{im} \leqslant Z_i, \quad i=1, \ldots, N, \quad m=1, \ldots, NM.$$

$$(ii) \quad Z_i \leqslant \bar{z}_i, \quad i=1, \ldots, N.$$

$$(iii) \quad \sum_{(i,m)\ on\ P} \frac{1}{z_{im} c_{im}} \leqslant T(P),$$

for all paths P from a starting node to a finishing node, where $T(P)$ is the project span for path P specified by the initial greedy policy.

$$(iv) \quad z_{im}, \ Z_i \geqslant 0, \quad i=1, \ldots, N, \quad m=1, \ldots, NM.$$

The above program may be reformulated by substituting time variables

$$t_{im} = \frac{1}{z_{im} c_{im}}, \quad i=1, \ldots, N, \quad m=1, \ldots, NM, \tag{2.37}$$

for intensities, and by making the substitution

$$B_i = \frac{1}{Z_i}, \quad i=1, \ldots, N, \tag{2.38}$$

as follows:

$$\textit{Minimize} \quad \sum_{i=1}^{N} \frac{p_i}{B_i}$$

subject to

$$S(1)\quad B_i - c_{im}t_{im} \leqslant 0,\ i=1,\ldots,N,\ m=1,\ldots,NM.$$

$$S(2)\quad c_{im}t_{im} \geqslant \frac{1}{\bar{z}_i},\ i=1,\ldots,N,\ m=1,\ldots,NM.$$

$$S(3)\quad \sum_{(i,m)\ on\ P} \frac{1}{z_{im}c_{im}} \leqslant T(P),$$

for all paths P from a starting node to a finishing node, where $T(P)$ is the project span for path P specified by the initial greedy policy.

$$S(4)\quad t_{im},\ B_i \geqslant 0.$$

We now have a convex objective to be minimized over a linear constraint set. Efficient algorithms exist for solving problems of this form, such as the Frank-Wolfe Linearization Algorithm. See references [1,4].

Many of the time assignments t_{im} may be critical in the initial greedy calculation, and so the smoothing program may be further simplified by eliminating such variables. A substitution of slack variables

$$s_{im} = t_{im} - \frac{1}{\bar{z}_i c_{im}},\ i=1,\ldots,N,\ m=1,\ldots,NM, \qquad (2.39)$$

is also efficient, eliminating the need for constraints $S(2)$. These steps effectively reduce the problem to consideration of a subnetwork of the extended network, namely the network of nodes which have slack for a greedy policy production plan. We shall refer to such a network as the "slack subnetwork". Constraints $S(3)$ would then effectively consider only slack subpaths of the extended network, ie, paths in the slack subnetwork.

In general, the program calculates a minimal set of activity intensity bounds

$$\left\{Z_i \mid Z_i = \frac{1}{B_i},\ i=1, \ldots, N\right\}$$

sufficient to meet the greedy output schedule. For our simple shipbuilding example, only the peak intensities of activities *B* and *D* can be reduced without extending the project span. The amount of reduction of each would, of course, depend on the relative magnitude of the smoothing weights chosen.

For further improvement in activity loading for the given resource preallocations, one must consider extensions of the project span. Indeed, a sensitivity analysis could be conducted by incrementing the project span length in constraints $S(3)$ for more reduction in loading peaks.

2.7. Resource Leveling

We have seen in section 2.6 how activity loading may be smoothed for a given resource preallocation. A smooth-loaded production plan offers technical improvement over the greedy plan in terms of the loading of machine services and other resources preallocated to each activity. We now consider the problem of improving the economic efficiency of the production system in terms of the *network* resource loading behavior.

It is important to distinguish two different categories of resources, *storable* and *non-storable.* Storable resources are materials "used up" in production, which can be stored from day to day before use at negligible costs. For such resources, we shall assume the behavior of loading histories is insignificant from a cost standpoint, so that only the total amounts used are of economic concern. In considering various time substitutions of the greedy production plan which meet the same project schedule, the integral of each resource history remains constant, and so no economic improvement is gained from time substituting demand for such resources.

Non-storable resources are the services per unit time of machines and labor. Such resources cannot be "hired and fired" with the ups and downs of system demands, and consequently resources adequate to meet peak demands must typically be maintained for the life of a construction project. Hence economic improvement of a production plan comes from time substitutions which reduce loading peaks for non-storable resources.

Such motivation for driving down resource peaks while attempting to avoid slippage of the project schedule has been well recognized. Many efforts on scheduling activities, ie, shifting activities between early and late start, to improve resource loading are contained in the literature. See references [3,6]. However, our approach here will be to consider the improvement possible by decreasing activity intensities (ie, increasing time assignments to the production of output units). For a node (i,m) of the extended network, the reduction in demand level for resource k resulting from increasing the time assignment t_{im} is given by

$$\frac{d}{dt_{im}}\left(\frac{a_{imk}}{t_{im}c_{im}}\right) = -\frac{a_{imk}}{t_{im}^2 c_{im}} \ . \qquad (2.40)$$

In the analysis of resource loading, trade-offs between resources often arise, whereby replanning to mitigate peaking behavior of one resource may transfer peaking problems to another resource. In the following analysis, peak costs for each non-storable resource are introduced to properly weigh such trade-offs.

2.7.1. Peak Pricing Model

Let $NS \leqslant NK$ be the number of non-storable resources, and renumber (if necessary) such resources $k=1, \ldots, NS$. Let p_k be the price per unit capacity of resource k, $k=1, \ldots, NS$. Each p_k represents the cost of maintaining a unit of capacity of resource k (eg, the salary of a laborer) for the life of a project, where such capacity is required to meet peak demands. We now consider the programming problem to accomplish a maximal amount of resource leveling (for these peak prices) of an initial greedy production plan, without violating production due dates. Let

$$\bar{A}_k, \ k=1, \ldots, NS,$$

represent the peak demand per time unit for exogenous resource k by the production system. For a production system in which each activity produces NM output units, the objective is to

$$\mathit{Minimize} \quad \sum_{k=1}^{NS} p_k \bar{A}_k$$

subject to

$$R(1) \quad \sum_{(i,m) \text{ on } P} t_{im} \leqslant T(P),$$

for all paths P from a starting node to a finishing node, where $T(P)$ is the project span for path P specified by the initial greedy policy.

$$R(2) \quad \sum_{m=1}^{NM} \sum_{i=1}^{N} \frac{a_{imk}}{t_{im} c_{im}} \delta(i,m,\tau) \leqslant \bar{A}_k,$$

$k=1, \ldots, NS,\ \tau=0, \ldots, T-1,$ *where*

$$\delta(i,m,\tau) = \begin{cases} 1 & \text{if } (i,m) \text{ } \mathit{operates} \\ & \mathit{during}\ [\tau, \tau+1) \\ 0 & \text{if } \mathit{not}. \end{cases} \tag{2.41}$$

$$R(3) \quad t_{im} \geqslant \frac{1}{\bar{z}_i c_{im}}, \quad i=1, \ldots, N, \quad m=1, \ldots, NM.$$

$$R(4) \quad t_{im},\ \bar{A}_k \geqslant 0, \quad i=1, \ldots, N, \quad m=1, \ldots, NM,$$

$k=1, \ldots, NK.$

Here constraints $R(1)$ insure no project slippage, where the time assignment to node (i,m) is t_{im} (ie, the intensity assignment to node (i,m) is $\frac{1}{t_{im} c_{im}}$). The constraints $R(2)$ define the resource peaks $\{\bar{A}_k\}$, $k=1, \ldots, NS$. The objective minimizes the costs of resource peaking.

Relating node start times to a time grid $\tau=0, 1, 2, \ldots, T-1$, as is required by constraints $R(2)$, is a difficult task. The best approach in the literature uses integer variables, making the problem too large to solve for all but the smallest time grids. (See references [3] and [6].) The approach taken here is to solve a sequence of subproblems approximating the exact formulation, in an effort to converge to an optimum.

Peak Pricing Procedure

In each subproblem, constraints approximating the constraints $R(2)$ are applied only to nodes which operate inside of the regions of peak resource demand for the solution at hand. The smaller, approximate problem is then optimized. Inspection is then made to see if peak regions have shifted, and, if so, to redefine the set of active constraints for the next subproblem. This process continues until no peak movement is observed.

Step 0. Preallocate resources and compute the greedy policy production plan.

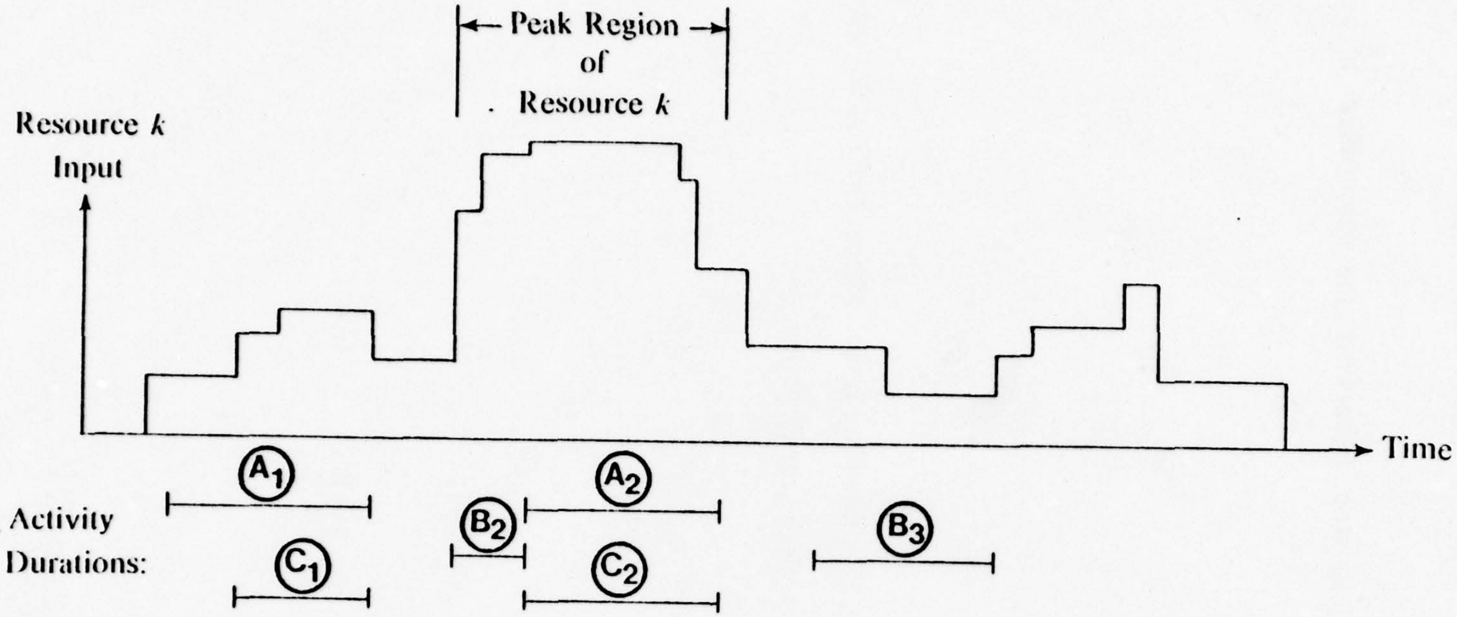
Step 1. From the resource histories of the *current solution*, *identify* peak time regions of each resource to be priced. The region specification is somewhat arbitrary, but should include peak and near-peak time points and exclude time regions of low resource demand. Regions not priced are likely to experience higher levels of resource utilization in the next solution. See Figure 2.7 for an example of region pricing.

Based on these region definitions, compute

$$\delta(i,m,k) = \begin{cases} 1 & \text{if } (i,m) \textit{ operates in a peak} \\ & \textit{region of resource } k \\ 0 & \text{if } \textit{not} \end{cases},$$

$$i=1,\ldots,N,\ m=1,\ldots,NM,\ k=1,\ldots,NS. \qquad (2.42)$$

If the



Pricing
Assignments: $\delta(A,1,k) = \delta(C,1,k) = \delta(B,3,k) = 0$

$\delta(A,2,k) = \delta(B,2,k) = \delta(C,2,k) = 1$

Figure 2.7
Example of Peak Region Pricing

$$\left\{\delta(i,m,k)\right\},\ i=1,\ldots,N,\ m=1,\ldots,NM,\ k=1,\ldots,NS,$$

have not changed since last iteration, stop; otherwise, the subproblem to be solved is

$$\textit{Minimize}\quad \sum_{k=1}^{NS} p_k \bar{A}_k$$

subject to

$$(i)\quad \sum_{(i,m)\ on\ P} t_{im} \leqslant T(P),$$

for all paths P from a starting node to a finishing node, where $T(P)$ is the project span for path P specified by the initial greedy policy.

$$(ii)\quad \sum_{m=1}^{NM} \sum_{i=1}^{N} \frac{a_{imk}}{t_{im} c_{im}}\ \delta(i,m,k) \leqslant \bar{A}_k,$$

$k=1,\ldots,NS.$

$$(iii)\quad t_{im} \geqslant \frac{1}{\bar{z}_i c_{im}},\ i=1,\ldots,N,\ m=1,\ldots,NM.$$

$$(iv)\quad t_{im},\ \bar{A}_k \geqslant 0,\ i=1,\ldots,N,\ m=1,\ldots,NM,$$

$k=1,\ldots,NS.$

Equivalently, we may solve

$$\textit{Minimize}\quad \sum_{k=1}^{NS} \sum_{i=1}^{N} \sum_{m=1}^{NM} p_k \frac{a_{imk}}{c_{im}} \frac{1}{(s_{im}+L_{im})}\ \delta(i,m,k)$$

subject to

$$M(1) \quad \sum_{(i,m) \text{ on } P} s_{im} \leqslant T(P) - \sum_{(i,m) \text{ on } P} L_{im},$$

for all paths P from a starting node to a finishing node, where $T(P)$ is the project span for path P specified by the initial greedy policy, where

$$L_{im} = \left[\Xi_i c_{im}\right]^{-1}, \; i=1, \ldots, N, \; m=1, \ldots, NM, \qquad (2.43)$$

are the lower time limits, and

$$s_{im} = t_{im} - L_{im}, \; i=1, \ldots, N, \; m=1, \ldots NM, \qquad (2.44)$$

are the slack variables.

$$M(2) \; s_{im} \geqslant 0, \; i=1, \ldots, N, \; m=1, \ldots, NM.$$

The subproblem has been converted to a nonlinear objective subject to linear constraints, and so may be readily optimized by algorithms such as Frank-Wolfe. See references [1,4].

Step 2. Set $t_{im} = s_{im} + L_{im}$, and go to Step 1.

At each iteration, resource peaks are "priced" down, whereby peak heights are reduced, but often peaks are shifted in location. The pricing scheme

$$\left\{\delta(i,m,k)\right\}, \; i=1, \ldots, N, \; m=1, \ldots, NM, \; k=1, \ldots, NS,$$

is then changed to reflect new peak regions for another iteration. When peaks no longer shift, the procedure terminates.

In each iteration, optimizing the subproblem makes all nodes in the extended network critical. This can be seen by noting that any slack on a start-to-finish path could be allocated to those activities operating in a peak resource region, thus decreasing the objective. Thus the procedure looks only among solutions in which all activities are critical *all the time.*

The procedure is similar to a relaxation method in optimization, as the constraints approximating $R(2)$ are applied only to nodes for which such constraints are "tight" in the current solution. Relaxation methods of optimization are formally guaranteed to converge to an optimum only when convexity of constraints is assured. This is not the case here, and it is not difficult to construct a simple example which will cause the procedure to cycle two solutions. However, the pricing policy may be readily adapted to allow the immediate escape from such a cycle, by simply pricing peak regions from both solutions.

Activity loading may not necessarily be smooth from resource leveling. However, loading for a given activity A_i can be made smoother by introducing a high price p_k on a real or fictitious resource k that is used by A_i and not shared with other activities. In the special case that each resource is unique to some activity, the resource leveling problem reduces to the smooth loading problem discussed in section 2.6. Further resource leveling requires a longer project span, and it may be useful to study the trade-off of resource peaks with project time length. The mathematical formulation can be easily modified to address such a problem by introducing a cost rate γ per unit time on project length. For simplicity of exposition, let us consider a production system for which the extended network has only one starting node and one finishing node, and for which the time T required to complete NM output units is the length to which an opportunity cost γ would apply.

The Peak Pricing Procedure can be applied to this problem by modifying the subproblem of Step 1 to

$$Minimize \quad \sum_{k=1}^{NS} \sum_{i=1}^{N} \sum_{m=1}^{NM} p_k \frac{a_{imk}}{c_{im}} \frac{1}{(s_{im}+L_{im})} \delta(i,m,k) + \gamma T$$

subject to

$$M(1)' \quad \sum_{(i,m)\ on\ P} s_{im} \leqslant T - \sum_{(i,m)\ on\ P} L_{im},$$

for all paths P from start to finish.

$$M(2)' \quad s_{im} \geqslant 0,\ i=1,\ldots,N,\ m=1,\ldots,NM,\ T \geqslant 0.$$

Note that only one additional variable occurs in the modified formulation, and the remainder of the procedure could be applied as before.

The comments in section 2.6 regarding reduction of the smooth loading program to consideration of only the slack subnetwork for the greedy production plan apply as well to the Peak Pricing Model. The subproblem of Step 1 of the Procedure can be effectively restricted to nodes which initially have slack, as time and intensity assignments for other nodes are fixed. Of course, the resource loads for the entire network must be used to determine peak regions.

2.7.2. Application to Ship Overhaul

The Peak Pricing Procedure was applied to a network consisting of 151 activities modeling a ship overhaul. Exogenous inputs to the activities consisted of labor hours of twelve different craft shops, such as pipefitting, welding, electrical work, etc. For such inputs, a reduction in resource peaks was equivalent to a reduction in staffing requirements for the overhaul.

Although the overhaul was a single output construction project, the concept of activity intensity adjustment for resource leveling could be applied. As only a single ship was being overhauled, the activity network would serve as the extended network for application of the model.

Shipyard data on estimated man-hour effort from each shop and estimated durations for the activities were available. From these data, intensity measurements and technical coefficients were developed as follows. Among the various shops, the shop with the greatest man-hour effort in a given activity A_i was selected to be the "lead shop" for A_i, whereby the intensity $z_i(t)$ of A_i on day t would equal the lead shop man-hours applied on day t.

The output coefficient c_i for a given activity A_i was taken to be

$$c_i = \left[\textit{total lead shop man-hours needed to finish } A_i \right]^{-1},$$

so that

$$z_i(t)c_i = (\textit{fraction of } A_i \textit{ completed on day } t).$$

The assumption was made that application of man-hours from other shops is proportional to lead shop effort, so that input coefficients were taken to be simple ratios of total activity effort, ie,

$$a_{ik} = \frac{(\textit{total man-hours of shop } k \textit{ needed to finish } A_i)}{(\textit{total lead shop man-hours needed to finish } A_i)},$$

which implies

$$z_i(t)a_{ik} = (\textit{man-hours of shop } k \textit{ applied to } A_i \textit{ on day } t).$$

Since the arcs of the activity network represented solely critical path-style precedences, the product transfer coefficients were simply defined by

$$\bar{a}_{ji} = \begin{cases} 1 \text{ if } D(i,j)=1 \\ 0 \text{ if } \textit{not}. \end{cases}$$

It was assumed that activity intensities could not exceed levels implied by the yard's estimated durations, but that activities could operate at lower intensities, ie, require longer durations. The intensity bound for each activity A_i was thus calculated to be

$$\bar{z}_i = (c_i)^{-1}(\textit{estimated duration of } A_i)^{-1} =$$

$$\frac{(\textit{total lead shop man-hours needed to finish } A_i)}{(\textit{estimated duration of } A_i)} .$$

As discussed earlier, trade-offs between the various resources are a key feature of resource leveling, whereby replanning to reduce peaking behavior in one shop may increase peaking problems in another shop. To value the relative expense of load peaks, capacity prices p_k were developed. These prices were chosen to be the sum of average wage and overhead rates per man-hour for each shop. In this way, more incentive was provided to level-load more "expensive" shops.

FORTRAN IV computer codes to calculate greedy policies and to minimize the nonlinear program in each iteration of the Peak Pricing Procedure were applied to these data. An initial forward greedy policy run was made to predict shop loads for an early start schedule based on the shipyard's planned activity durations. (Resource preallocations allowing activity operation at intensity bounds were assumed.) This run established a project time span of 370 working days. The Peak Pricing Procedure could have been immediately applied to the slack subnetwork, but instead, some observations about the nature of the overhaul served to reduce the problem size.

As is typical of construction projects, resource loads were high in the middle of the project duration, but low near the beginning and end of the project. Activities near the start or finish of the overhaul consisted largely of tests and inspections, which, while requiring time to perform, were not labor-intensive. It was clear that replanning could not shift shop workload peaks to the very early or late stages of the project, and that operating activities during these periods at lower intensities could not serve to reduce workload peaks.

For this reason, the slack subnetwork under consideration was reduced to that consisting of activities whose start times and slack possibly allowed them to operate in the middle range of the project where resource peaks could occur. The reduced network for application of the Peak Pricing Procedure then consisted of 33 activities contained in 48 slack paths.

The computer code used in the Peak Pricing Procedure applied the Frank-Wolfe linearization algorithm, which computed a solution within a 1% tolerance of the optimum for each subproblem. (See reference [1].) Peak resource regions were defined to be all work days where resource demand was within 4% of peak demand.

After four iterations, peak regions stabilized, so that no further improvement was possible for the Procedure. Each iteration had required about 10-15 seconds effective time on a CDC 6400 computer. Figure 2.8 compares work loads of the mechanical shop for the greedy production plan and the resource-leveled production plan. As can be seen, considerable improvement in shop loading has been made. Overall, a 38% improvement in the objective function was made between the initial greedy run and the last iteration of the pricing procedure. In view of the man-hours involved in the project (on the order of 100,000), such leveling was significant.

Considerable replanning of activities to lower intensity levels was made by the Pricing Procedure to obtain this improvement. Undoubtedly, there are practical lower limits on activity intensity for many of the activities, for such an overhaul. However, such limitations could be easily incorporated into the

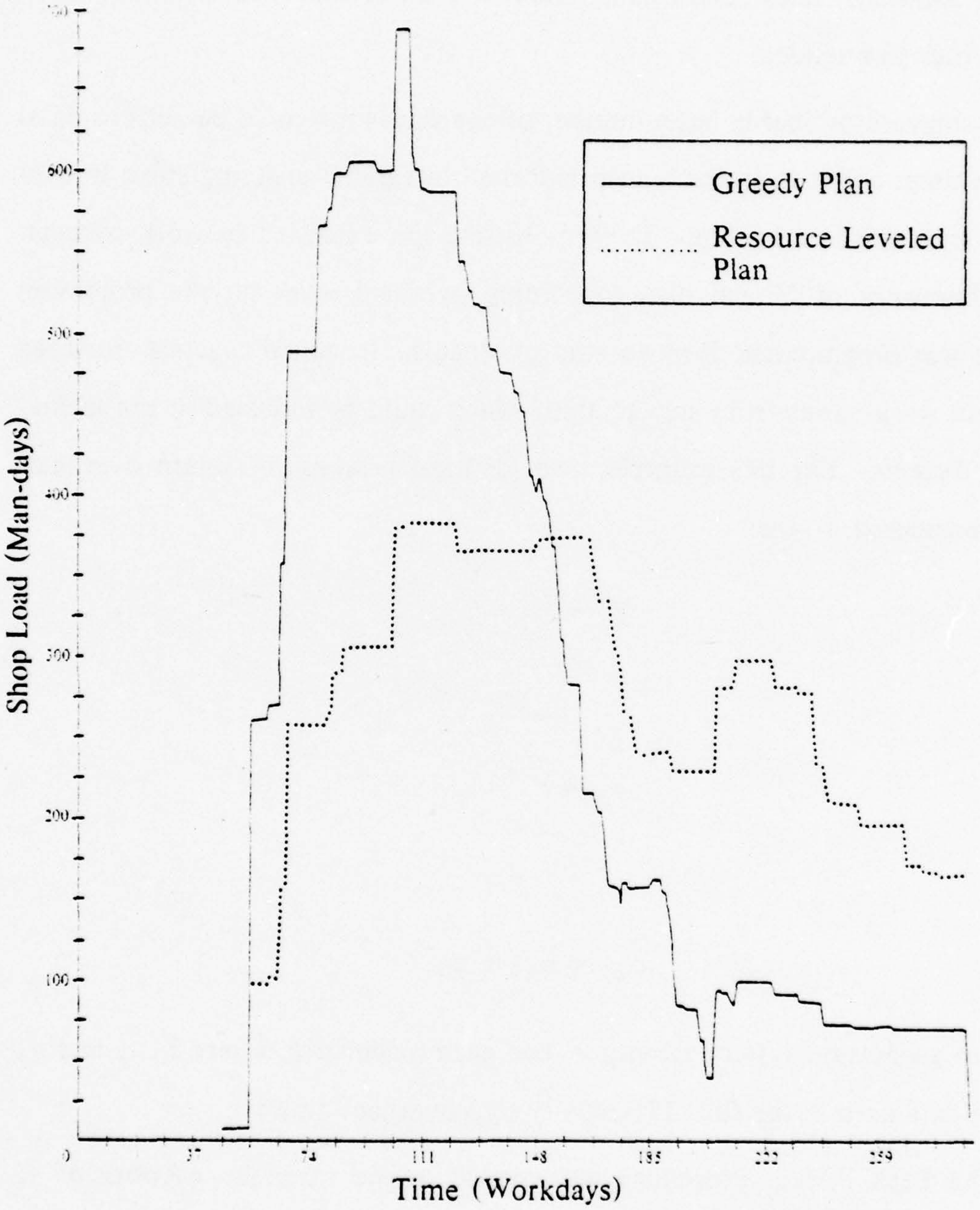


Figure 2.8
Mechanical Shop Work Load
Before and After Resource Leveling

model in the form of lower bounds (upper bounds) on activity intensity (duration). The constraints $M(1)-M(2)$ for the Peak Pricing Procedure would only require additional linear constraints expressing upper limits on slack allocation to activities so bounded.

A shipyard ordinarily has a number of overhaul projects to be performed at a given time, and it is desirable to perform an integrated planning effort to control overall resource loading. To demonstrate the extended network concept, the subnetwork of 21 activities concerning overhaul work on the propulsion system was programmed for two ship overhauls. In actual practice, required overhaul work varies from ship to ship, which could be reflected in the technical coefficients. For this example, two identical propulsion system overhauls were considered, so that

$$\Xi_{i1} = \Xi_{i2} = \Xi_i,$$

and

$$c_{i1} = c_{i2} = c_i,$$

and

$$a_{ik1} = a_{ik2} = a_{ik},$$

for each propulsion system activity A_i and each resource k, where Ξ_i, c_i and a_{ik} are the data used in the full, 151-activity ship overhaul network.

The Peak Pricing Procedure was applied to the extended network of 42 nodes for this problem. Figure 2.9 compares workloads for the shipboard mechanical shop for the greedy production plan and the resource leveled production plan. This result is typical of the smoothing obtainable when two ships under construction interface. Production planning which is done merely by

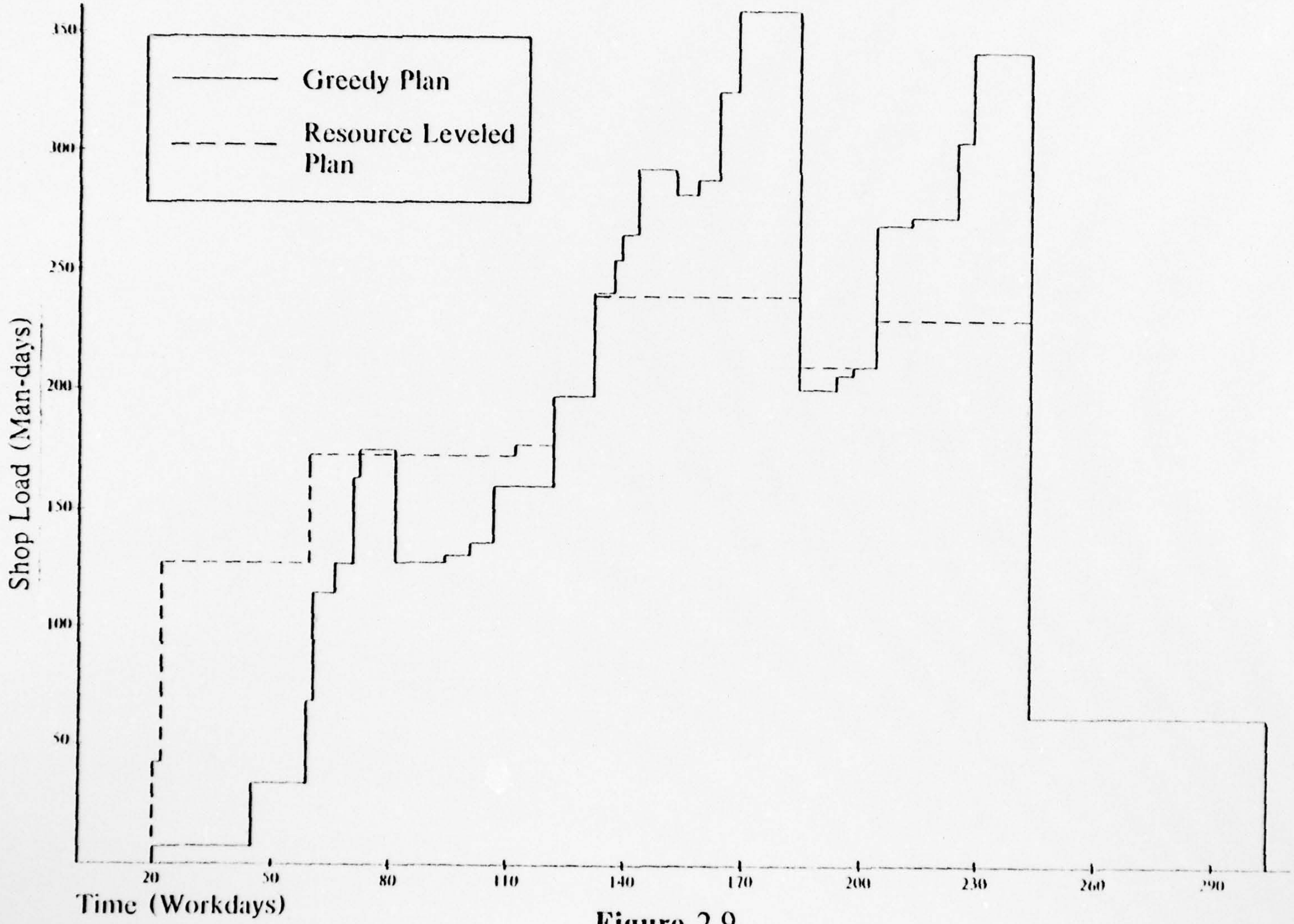


Figure 2.9
Shipboard Mechanical Shop Work Load – Two Ships

overlay of a plan for a single ship is bound to involve the kind of unsmoothed peaks shown.

3. Dynamic Production Planning With Continuous Flow Transfers of Intermediate Product

We have seen how the Discrete Transfer Model provides considerable detail concerning discrete interactions between production activities. However, providing such detail necessitates considerable computational complexity. Such calculational difficulties can be reduced by considering a limiting form of the Discrete Transfer Model, in which intermediate product transfers are linearly proportional to activity intensity. As will be shown, the same form of linear inequalities as in the Discrete Transfer Model governs this Continuous Flow Model. The linear inequalities have been adapted from those presented in reference [10], and have been extended to include product inventories and classes of exogenous input. It will also be shown that treating product transfers as continuous flows allows further development of the model, including allocation of resources by linear programming.

3.1. Development of Continuous Flow Model

For the Continuous Flow Model, we shall initially assume that each activity produces a single product, to be shared as intermediate product input by follow-on activities, and/or transferred as final product. The case where an activity produces a distinct product for each follow-on activity, as with critical path-style precedences, will also be discussed. It will also be assumed initially that no two activities produce the same intermediate product. As in the Discrete Transfer Model, inventories of intermediate product awaiting transfer shall be subject to storage capacities.

The activity network will now be allowed to contain cycles, representing more complicated activity dependencies. Leontief-like inter-industry flows can then be modeled dynamically, as well as such phenomena as recycling.

Resources for exogenous input are divided into two categories, *storable* and *non-storable.* Storable resources would include materials such as steel, lumber, fuel, etc, while non-storable resources would include services such as various kinds of labor hours or machine hours. It should be noted that limitations in the availability of storable and non-storable resources constrain a production system in different ways.

Non-storable resource limitations constrain the sum of activity resource demands at each time period. These limitations constitute capacity constraints. Storable resource limitations, however, constitute inventory constraints, whereby the sum of activity resource demands at each time period is limited by the available inventory of stored resources.

For a production system with a time horizon T, we shall use the following notation for the set of *technical limitations,* $L(T)$:

(a) $\left\{\bar{z}_i(t)\right\}_{i=1,\ldots,N}^{t=0,1,2,\ldots,T-1}$, the activity intensity bounds;

(b) $\left\{X_k(t)\right\}_{k=1,\ldots,NS}^{t=0,1,2,\ldots,T-1}$, the time histories of non-storable resource exogenous inputs, $NS \leqslant NK$;

(c) $\left\{Y_k(t)\right\}_{k=NS+1,\ldots,NK}^{t=0,1,\ldots,T-1}$, the time histories of storable resouce exogenous inputs, where

$$\sum_{\tau=0}^{t} Y_k(\tau)$$

is the cumulative amount of resource k supplied during $[0,t)$;

(d) $\left\{inv_i^{0}\right\}_{i=1,\ldots,N}$, the initial inventories of activity product (perhaps all zero) for intermediate uses; and

(e) $\left\{cap_i(t)\right\}_{i=1,\ldots,N}^{t=0,1,\ldots,T-1}$, the capacities for storage of activity product.*

For convenience, the intensity $z_i(t)$ of activity A_i, $i=1,\ldots,N$, on each time interval $[t,t+1)$, $t=0,1,2,\ldots,T-1$, shall be partitioned into effort producing intermediate product, $z_i^I(t)$, and effort producing final product, $z_i^F(t)$, where

$$z_i^I(t) + z_i^F(t) = z_i(t). \tag{3.1}$$

Of course, $z_i^I(t)$ or $z_i^F(t)$ may be zero for all t should activity A_i produce only final or intermediate product, respectively. With this notation, a production plan

$$\left\{z_i^I(t),\ z_i^F(t)\right\}_{i=1,\ldots,N}^{t=0,\ldots,T-1}$$

is said to be *feasible for L(T)* if the plan satisfies the following set of linear inequalities, $A(L(T))$:

A(L(T))1. $\sum_{i=1}^{N} a_{ik}(t)z_i(t) \leqslant X_k(t)$, $k=1,\ldots,NS$, $t=0,1,\ldots,T-1$.

A(L(T))2. $\sum_{\tau=0}^{t}\sum_{i=1}^{N} a_{ik}(\tau)z_i(\tau) \leqslant \sum_{\tau=0}^{t} Y_k(\tau)$, $k=NS+1,\ldots,NK$,

$t=0,\ldots,T-1$.

* If a product inventory is not capacity-constrained, the capacity may be set equal to some very large number for programming purposes.

A(L(T))3. $\sum_{\tau=1}^{t}\sum_{i=1}^{N}\bar{a}_{ij}(\tau)z_i(\tau) \leqslant inv_j^0 + \sum_{\tau=0}^{t-1}c_j(\tau+1)z_j^I(\tau),$

$j=1,\ldots,N,\ t=1,\ldots,T-1,$ *and*

$$\sum_{i=1}^{N}\bar{a}_{ij}(0)z_i(0) \leqslant inv_j^0,\ j=1,\ldots,N.$$

A(L(T))4. $inv_j^0 + \sum_{\tau=0}^{t-1}c_j(\tau+1)z_j^I(\tau) - \sum_{\tau=1}^{t}\sum_{i=1}^{N}\bar{a}_{ij}(\tau)z_i(\tau) \leqslant cap_j(t),$

$j=1,\ldots,N,\ t=1,\ldots,T-1,$ *and*

$$inv_j^0 - \sum_{i=1}^{N}\bar{a}_{ij}(0)z_i(0) \leqslant cap_j(0),\ j=1,\ldots,N.$$

A(L(T))5. $z_j^I(t) + z_j^F(t) = z_j(t),\ j=1,\ldots,N,\ t=0,\ldots,T-1,$

$$z_j(t) \leqslant \bar{z}_j(t),\ j=1,\ldots,N,\ t=0,\ldots,T-1,$$

$$z_j^I(t),\ z_j^F(t) \geqslant 0,\ j=1,\ldots,N,\ t=0,\ldots,T-1.$$

Constraints A(L(T))1 and A(L(T))2 express resource limitations. Constraints A(L(T))3 insure adequate intermediate product input exists to support production activity, while constraints A(L(T))4 insure that inventories of intermediate products do not exceed capacities. Finally, constraints A(L(T))5 limit intensities to non-negative values less than intensity bounds.

In the case an activity A_α produces a distinct product for each follow-on activity, the constraint set $A(L(T))$ is modified as follows. Let

$$inv_{\alpha,i}(t)$$

denote the inventory of product produced by A_α for follow-on activity A_i, $i \neq \alpha$, at the time t. Similarly, let

$$cap_{\alpha,i}(t)$$

denote the capacity of such inventory. Then for $j=\alpha$, constraints **A(L(T))3** and **A(L(T))4** are revised to the following:

$$\textbf{A(L(T))3}'. \quad \sum_{\tau=1}^{t} \bar{a}_{i\alpha}(\tau) z_i(\tau) \leqslant inv^0_{\alpha,i} + \sum_{\tau=0}^{t-1} c_\alpha(\tau+1) z^I_\alpha(\tau),$$

$i \neq \alpha$, $t=1, \ldots, T-1$, *and*

$$\bar{a}_{i\alpha}(0) z_i(0) \leqslant inv^0_{\alpha,i}, \quad i \neq \alpha.$$

$$\textbf{A(L(T))4}'. \quad inv^0_{\alpha,i} + \sum_{\tau=0}^{t-1} c_\alpha(\tau+1) z^I_\alpha(\tau) - \sum_{\tau=1}^{t} \bar{a}_{i\alpha}(\tau) z_i(\tau) \leqslant cap_{\alpha,i}(t),$$

$i \neq \alpha$, $t=1, \ldots, T-1$, *and*

$$inv^0_{\alpha,i} - \bar{a}_{i\alpha}(0) z_i(0) \leqslant cap_{\alpha,i}(0), \quad i \neq \alpha.$$

In the case two activities produce the same intermediate product, say A_α and $A_{\alpha+1}$, the constraint set $A(L(T))$ can also be easily modified. For this case we shall assume the combined storage capacity

$$cap_\alpha(t) + cap_{\alpha+1}(t)$$

is available for storage of such product. Then for $j=\alpha$ and $j=\alpha+1$, constraints **A(L(T))3** and **A(L(T))4** are revised to the following:

A(L(T))3″. $\sum_{\tau=1}^{t} \sum_{i=1}^{N} \bar{a}_{i\alpha}(\tau) z_i(\tau) \leqslant \sum_{j=\alpha}^{\alpha+1} inv_j^0 + \sum_{\tau=0}^{t-1} \sum_{j=\alpha}^{\alpha+1} c_j(\tau+1) z_j^f(\tau),$

$t=1, \ldots, T-1,$ *and*

$$\sum_{i=1}^{N} \bar{a}_{i\alpha}(0) z_i(0) \leqslant \sum_{j=\alpha}^{\alpha+1} inv_j^0.$$

A(L(T))4″. $\sum_{j=\alpha}^{\alpha+1} inv_j^0 + \sum_{\tau=0}^{t-1} \sum_{j=\alpha}^{\alpha+1} c_j(\tau+1) z_j^f(\tau) - \sum_{\tau=1}^{t} \sum_{i=1}^{N} \bar{a}_{ij}(\tau) z_i(\tau) \leqslant$

$\sum_{j=\alpha}^{\alpha+1} cap_j(t), \ t=1, \ldots, T-1,$ *and*

$$\sum_{j=\alpha}^{\alpha+1} inv_j^0 - \sum_{i=1}^{N} \bar{a}_{i\alpha}(0) z_i(0) \leqslant \sum_{j=\alpha}^{\alpha+1} cap_j(0).$$

Note that it is required that $\bar{a}_{i,\alpha}(t)$ and $\bar{a}_{i,\alpha+1}(t)$ be identical, because the same product is produced by A_α and $A_{\alpha+1}$.

In any case, $A(L(T))$ constitutes a set of linear inequalities. For simplicity of exposition, we shall assume in what follows that each activity produces a single product and that no two activities produce the same intermediate product.

It should be noted that in the case the activity network includes cycles, there must be non-zero initial inventories of intermediate products for input to the activities comprising such cycles. Otherwise, constraints A(L(T))3 would force zero intensities for these activities for all time. The relationship between the magnitude of the initial inventories

$$\left\{ inv_i^0 \right\}_{i=1,\ldots,N}$$

and the final output realizable from the network is discussed in section 3.4.

Although not required, resources could be preallocated through time among the activities as in the Discrete Transfer Model, and simulation of the production process could be performed using greedy policies to determine the resulting maximum throughput, or minimum project span to reach a given output accumulation. In reference [10], greedy policy algorithms for the Continuous Flow Model were developed as part of an effort to demonstrate the complexity of dynamic correspondences and the dependency upon resource assignments. (These algorithms require a cycle-free network.) However, our approach here will be to consider the allocation of resources by linear programming.

3.2. Output Maximization

Assume the production system can produce $P \geqslant 1$ final products. A specific product mix may be desired, and we shall consider the problem of maximizing the scale of this product mix accumulated by a time horizon T.

For this purpose, let z_{N+1} be a variable indexing the amount of the product mix accumulated by time T. The amounts of the various products will be related by coefficients

$$\bar{a}_{N+1,p}, \; p=1, \ldots, P,$$

where $\bar{a}_{N+1,p}z_{N+1}$ is the amount of product p accumulated. These coefficients specify the product mix to be produced. For convenience, we define constants

$$\Delta(i,p) = \begin{cases} 1 \text{ if } A_i \textit{ produces} \\ \quad \textit{product } p \\ 0 \text{ if } \textit{not} \end{cases}, \; i=1, \ldots, N, \; p=1, \ldots, P, \tag{3.2}$$

indicating which activities produce each final product.

The maximum accumulated output of a specified product mix at time T from production activity during $[0,T)$ is given by the optimum of the following linear program.

Maximize z_{N+1}

subject to

$$(i) \quad \bar{a}_{N+1,p}z_{N+1} \leqslant \sum_{t=0}^{T-1} \sum_{i=1}^{N} \Delta(i,p)c_i(t+1)z_i^F(t), \; p=1, \ldots, P.$$

$$(ii) \quad \left\{ z_i^I(t), \; z_i^F(t) \right\}_{i=1, \ldots, N}^{t=0, \ldots, T-1} \in A(L(T)),$$

and $z_{N+1} \geqslant 0$.

In general, the program involves $(N)(T) + 1$ variables (plus slack variables), with up to

$$(NK)(T) + 3(N)(T) + P$$

constraints. Clearly, the time horizon (ie, the number of time grid periods) is the most sensitive factor in terms of the problem size which can be handled. The constraint set $A(L(T))$, consisting of a mixture of inventory balance equations, and capacity and resource allocation constraints, has a structure which can be exploited as follows.

Note that the slack variables for constraints A(L(T))3 represent the inventories of intermediate product stored during the intervals $[t,t+1)$, $t=0, \ldots, T-1$. Calling these variables

$$\left\{ inv_j^I(t) \right\}, \; t=0, \ldots, T-1,$$

we may rewrite constraints A(L(T))3 and A(L(T))4 as follows:

A(L(T))3. $\sum_{i=1}^{N} \bar{a}_{ij}(t) z_i(t) - c_j(t) z_j(t-1) - inv_j^I(t-1) + inv_j^I(t) = 0,$

$j=1, \ldots, N,\; t=1, \ldots, T-1,$ *and*

$$\sum_{i=1}^{N} \bar{a}_{ij}(0) z_i(0) + inv_j^I(0) = inv_j^0, \; j=1, \ldots, N.$$

A(L(T))4. $inv_j^I(t) \leqslant cap_j(t), \; j=1, \ldots, N, \; t=0, \ldots, T-1.$

With this rewriting of the constraints, it is evident that the constraints A(L(T))1, A(L(T))4 and A(L(T))5 apply only time unit by time unit, while constraints A(L(T))2 and A(L(T))3 link variables through time. Hence the constraint matrix exhibits partial block diagonal structure, as shown in Figure 3.1. As can be seen,

$$2(N)(T) + (NS)(T)$$

constraints comprise the block diagonal structure, with

$$(N)(T) + (NK - NS)(T) + P$$

linking constraints. Good potential is thus offered for application of an efficient large-scale programming procedure, such as decomposition. (See reference [4].) Using decomposition, at each iteration, up to T independent subprograms, each with $(2N + NS)$ constraints, would be solved. In the case intensity bounds, resource and inventory capacities, and technical coefficients are constant in time, these subprograms would have identical constraint matrices.

As opposed to maximizing the throughput of a given output mix, one could consider maximizing the value of output produced. We suppose each product p has a unit price v_p which indicates the value added in manufacture, or sales price, etc. Then the maximum value of output accumulated from production activity during $[0, T)$ is given by the optimum of the following linear program:

$$\textit{Maximize} \sum_{p=1}^{P} \sum_{t=0}^{T-1} \sum_{i=1}^{N} v_p \Delta(i,p) c_i(t+1) z_i^F(t)$$

subject to

$$\left\{ z_i^I(t),\ z_i^F(t) \right\}_{i=1,\ldots,N}^{t=0,\ldots,T-1} \in A(L(T)).$$

The remarks about problem size and structure concerning the previous program

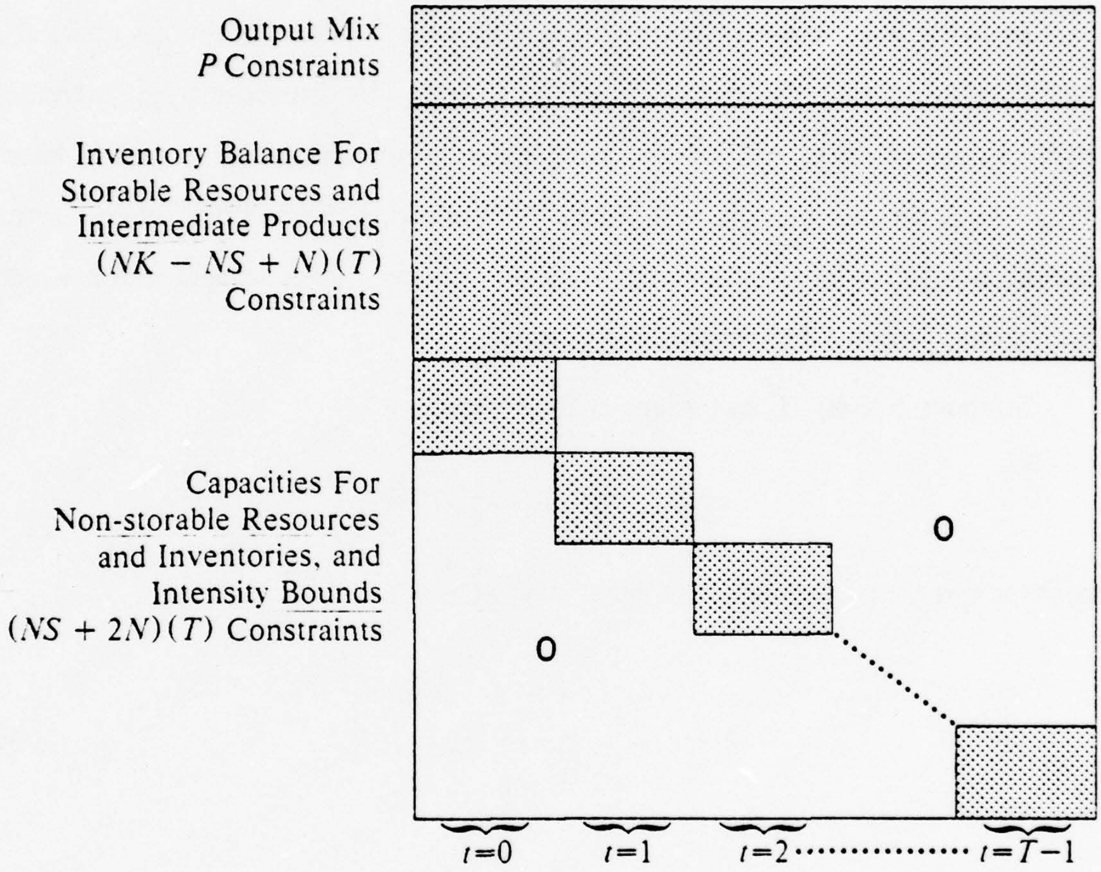


Figure 3.1
Structure of Constraint Matrix For Output Maximization

apply here as well, as only the product mix variable and its P constraints have been deleted.

Modification for Intensity Lower Bounds

In many production systems, indivisibilities and other conditions make the operation of certain activities infeasible at very low intensities, ie, activities must either be "shut off" or operating above some minimum intensity level. While such constraints are non-linear, the following formulation converts the output maximization problem to a mixed integer, linear program for which branch and bound solution procedures exist.

Suppose activity A_j has intensity lower bounds

$$\underline{z}_j(t),\ t=0,1,\ldots,T-1,$$

perhaps constant. Define a new binary variable

$$\delta_j(t) = \begin{cases} 1 \text{ if } A_j \textit{ is operating} \\ \quad \textit{during } [t,t+1) \\ 0 \text{ if } \textit{not.} \end{cases} \tag{3.3}$$

The following constraints are then added to the formulation:

$$z_j(t) - \bar{z}_j(t)\delta_{j(t)} \leqslant 0,\ t=0,1,\ldots,T-1, \tag{3.4}$$

$$-z_j(t) + \underline{z}_j(t) - \bar{z}_j(t)\Big[1 - \delta_j(t)\Big] \leqslant 0,\ t=0,\ldots,T-1, \tag{3.5}$$

$$\delta_j(t) \leqslant 1,\ \textit{integer.} \tag{3.6}$$

This constraint formulation implies

$$\delta_j(t) = 0 \textit{ iff } z_j(t) = 0, \textit{ and}$$

$$\delta_j(t) = 1 \; iff \; z_j(t) \geqslant \underline{z}_j(t).$$

For each activity with lower bound constraints on intensity, $2(T)$ added constraints and T binary variables are introduced into the formulation. As before, the density of the time grid limits the size of the problem which can be solved when such constraints are present.

3.3. Cost Minimization

In section 3.2, the forward-looking problem of maximizing the throughput of production on a fixed time interval was examined. In this section, the backward-looking problem of determining a minimum cost production plan to meet a given demand schedule for output will be investigated.

It will be convenient to express output requirements for each product type in cumulative terms. Let

$$\hat{U}_p(t), \quad t=1,2,\ldots,T, \quad p=1,\ldots,P,$$

denote the required cumulative production of product p by time t. A demand schedule for the production system implies constraints

$$\sum_{\tau=0}^{t-1} \sum_{i=1}^{N} \Delta(i,p) c_i(\tau+1) z_i^F(\tau) \geqslant \hat{U}_p(t), \quad p=1,\ldots,P,$$

$$t=1,\ldots,T. \tag{3.7}$$

Here we allow final products to be produced in advance of required schedule and stored as inventory, although such inventories will be subject to holding costs, as discussed below.

Cost Formulation

The objective of the formulation will be to minimize costs for storable and non-storable resources, and for intermediate and final inventories, subject to meeting the demand schedule. Each of the cost categories is discussed below.

Non-storable resources have *capacity costs* corresponding to the peak demands for each such resource. As these resources cannot be accumulated, the production system must have the capability to accommodate peak loads.

Storable resources, however, have *prices.* Unless purchasing or delivery constraints exist, only the cumulative demand for such resources determine costs for such resources. If input coefficients are constant or vary only with cumulative activity output (eg, learning curve effects), costs for such resources are constant over all production plans producing the same total amounts of final products.

Inventories of intermediate product may also have capacity costs corresponding to the peak requirements for storage of such products. Inventories of final output, however, should bear an opportunity cost to properly discourage holding of excess inventories of final products. In this way, the trade-off between efficient production loading, and the maintenance of final product inventories in line with product demand, can effectively be made.

Assuming linear capacity costs, the cost minimization problem can be formulated as a linear program as follows.

Let

$$\vec{C}^{x} = \left\{ C_{1}^{x}, \ldots, C_{NS}^{x} \right\},$$

be the vector of costs per unit capacity for non-storable resources; let

$$\vec{C}^{y} = \left\{ C_{NS+1}^{y}, \ldots, C_{N}^{y} \right\},$$

be the price vector for storable resources; let

$$\vec{C}^{z} = \left\{ C_{1}^{z}, \ldots, C_{N}^{z} \right\},$$

be the vector of costs per unit capacity for storage of intermediate products; and let

$$\vec{C}^{F} = \left\{ C_{1}^{F}, \ldots, C_{P}^{F} \right\},$$

be the holding costs for final products per unit time.

To serve as variables in the minimization, let

$$\overline{X} = \{X_1, \ldots, X_{NS}\},$$

denote the peak amounts of non-storable resources required per unit time; let

$$\overline{Y} = \{Y_{NS+1}, \ldots, Y_{NK}\},$$

denote the total amounts of storable resources required by the project; let

$$\overline{cap} = \{cap_1, \ldots, cap_N\},$$

denote the required intermediate product inventory capacities; and let

$$\overline{inv}^F(t) = \{inv_1^F(t), \ldots, inv_P^F(t)\},$$

denote the inventories of final products in excess of the demand schedule at time t, for $t=1, 2, \ldots, T$.

For given intensity bounds

$$\left\{\Xi_i(t)\right\}_{i=1,\ldots,N}^{t=0,\ldots,T-1}$$

and initial intermediate product inventories

$$\left\{inv_j^0\right\}_{i=1,\ldots,N},$$

the minimum cost production plan meeting the output schedule

$$\left\{\hat{U}_p(t)\right\}_{p=1,\ldots,P}^{t=1,\ldots,T}$$

is given by the optimum of the following linear program.

$$\mathit{Minimize}\quad \vec{C}^{x}\cdot\vec{X} + \vec{C}^{y}\cdot\vec{Y} + \vec{C}^{z}\cdot\vec{cap} + \sum_{t=1}^{T} \vec{C}^{F}\cdot\vec{inv}^{F}(t)$$

subject to

$$\text{C1.}\quad \sum_{\tau=0}^{t-1}\sum_{i=1}^{N} \Delta(i,p)c_i(\tau+1)z_i^F(\tau) - inv_p^F(t) = \hat{U}_p(t),$$

$p=1, \ldots, P,\ t=1, \ldots, T.$

$$\text{C2.}\quad \sum_{i=1}^{N} a_{ik}(t)z_i(t) \leqslant X_k,\ k=1, \ldots, NS,\ t=0,1, \ldots, T-1.$$

$$\text{C3.}\quad \sum_{t=0}^{T-1}\sum_{i=1}^{N} a_{ik}(t)z_i(t) = Y_k,\ k=NS+1, \ldots, NK.$$

$$\text{C4.}\quad \sum_{i=1}^{N} \bar{a}_{ij}(t)z_i(t) - c_j(t)z_j(t-1) - inv_j^I(t-1) + inv_j^I(t) = 0,$$

$j=1, \ldots, N,\ t=1, \ldots, T-1,$ *and*

$$\sum_{i=1}^{N} \bar{a}_{ij}(0)z_i(0) + inv_j^I(0) = inv_j^0,\ j=1, \ldots, N.$$

$$\text{C5.}\quad inv_j^I(t) \leqslant cap_j(t),\ j=1, \ldots, N,\ t=0, \ldots, T-1.$$

$$\text{C6.}\quad z_j^I(t) + z_j^F(t) = z_j(t),\ j=1, \ldots, N,\ t=0, \ldots, T-1,$$

$$z_j(t) \leqslant \bar{z}_j(t),\ j=1, \ldots, N,\ t=0, \ldots, T-1,$$

$$\text{C7.}\quad \overline{X} = \left[X_1, \ldots, X_{NS}\right] \geqslant 0,$$

$$\overline{Y} = \left[Y_{NS+1}, \ldots, Y_{NK}\right] \geqslant 0.$$

$$c\overline{a}p = \left[cap_1, \ldots, cap_N\right] \geqslant 0,$$

$$\vec{inv}^I(t) = \left[inv_1^I(t), \ldots, \vec{inv}_N^I(t)\right] \geqslant 0, \; t=0, \ldots, T-1,$$

$$\vec{inv}^F(t) = \left[inv_1^F(t), \ldots, inv_P^F(t)\right] \geqslant 0, \; t=1, \ldots, T,$$

$$z_j^I(t), \; z_j^F(t) \geqslant 0, \; j=1, \ldots, N, \; t=0, \ldots, T-1.$$

Constraints C1 define the inventories of final products, and constraints C2 and C3 define the peak levels of resource demand. Constraints C5 define the required inventory capacities. Constraints C4 and C6 deal with inventory balance and intensity bounds in the same manner as the treatment of the output maximization problem.

It should be noted that the left hand side of constraints C3 can directly replace Y_k, $k=NS+1, \ldots, NK$, in the objective function, and so these constraints would not appear in any actual programming effort. Also, the intensities

$$\left\{z_i(t)\right\}_{i=1, \ldots, N}^{t=0, \ldots, T-1}$$

would be replaced in all constraints by the sum of intensity effort for intermediate and final product, ie,

$$z_i^I(t) + z_i^F(t) = z_i(t), \; i=1, \ldots, N, \; t=0, \ldots, T-1. \qquad (3.8)$$

In general, the program includes up to

$$\left[P + NS + 3N\right](T)$$

constraints. As before, the fineness of the time grid is the most sensitive factor in terms of the problem size which can in practice be solved. Figure 3.2 displays the structure of the constraint matrix. As can be seen, a bordered angular configuration is now exhibited. Although this is a more difficult structure than that for the output maximization problem, nonetheless it can be exploited. (See for example the discussion of Ritter's partitioning procedure in reference [4].)

The above formulation treats a "long run" problem, in a sense, because the various resource and inventory capacities are considered to be variables. A "short run" problem, in which some of these capacities were given data, would involve fewer variables, but still the same number of constraints. The degree of difficulty is thus roughly equivalent to that of the "long run" problem.

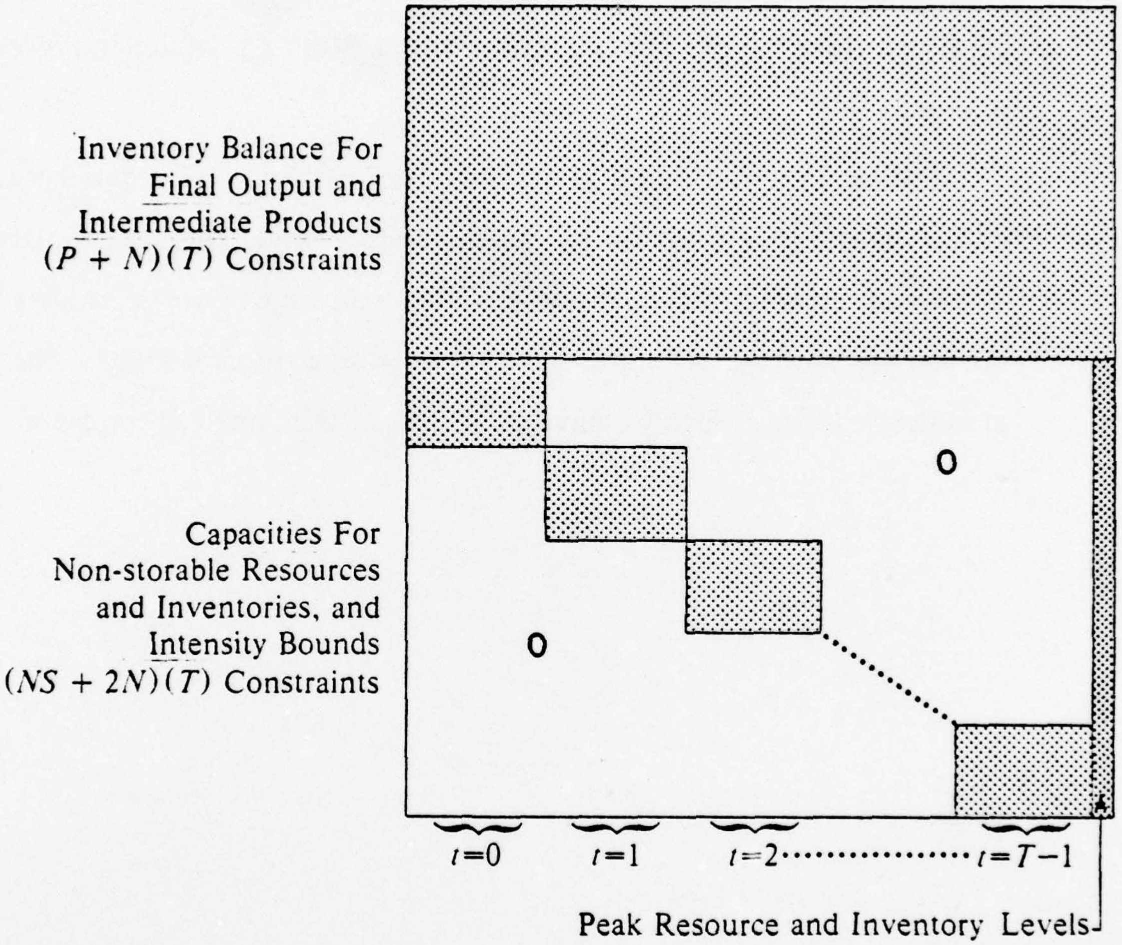


Figure 3.2
Structure of Constraint Matrix For Cost Minimization

3.4. Capacity and Efficiency of a Production System

The output maximizing procedure discussed in section 3.2 calculates the maximum level of output for a specific product mix (or the maximum value of output) producible in a given time span of production. We can thus speak of a production system *capacity* to produce a specific product mix (or maximum value of products) in a given time interval.

The marginal output return from increasing a resource or inventory capacity can be studied using parametric linear programming. (See reference [2].) Using a parametric right hand side on the output maximization linear program, it can be observed which inventory or resource capacities, initial inventories, or intensity bounds, if increased, would increase system capacity, and by how much. In general, the system capacity is a piecewise linear, concave function of each such technical limitation.

As mentioned in section 3.1, when the activity network includes cycles, some of the initial inventories of intermediate product must be non-zero to support production activity by activities comprising such cycles. The parametric programming effort described above will indicate what initial inventories are essential for system output.

The optimal tableau of the output maximizing program shows which resource and inventory capacities are fully utilized, and which are not. A resource fully utilized up to capacity can be said to be efficiently used, since a reduction in capacity would diminish output produced in the given time span. On the other hand, the tableau shows the amount of slack in resource constraints not "tight". Committing the full amount of such resources to production activity results in no more final output than if capacities were reduced by the values of the slacks.

For the given time span and product mix, an *efficient* set of resource and inventory capacities is obtained from the optimal tableau, namely, the reduction of initial capacities by the value of the slacks. This is *technical efficiency*, in the sense that a reduction in the availability of a resource (without some offsetting increase in other resources), would diminish output producible in the given span.

The cost minimization program, on the other hand, determines resource levels and inventory capacities which are *economically efficient* for a given schedule of final products to be delivered. If the given time length and the optimal output level from the output maximization program are used as the demand schedule constraint for the cost minimization program, an interesting comparison can be made. The levels of resource demand calculated by the latter program, which are *economically* efficient, could be contrasted with resource demands which were calculated to be *technically* efficient by the former program. In general, these demand levels will differ, because technically efficient levels are dependent on initial system capacities.

These concepts of efficiency and capacity are dependent on the time horizon T allowed for production, and the time variability of input histories comprising the set $L(T)$ of technical limitations. In order to provide a connection with steady state models of production, we shall consider the behavior of a single-output production system in the case input histories, technical coefficients and intensity bounds are constant through time.

Using the notation of section 3.2, let

$$\Delta(i,1) = \begin{cases} 1 \text{ if } \textit{activity } A_i \textit{ produces} \\ \quad \textit{final product} \\ 0 \text{ if } \textit{not} \end{cases}, \quad i=1,\ldots,N. \tag{3.9}$$

The output maximization program for a time horizon T is then

$$Maximize \sum_{t=0}^{T-1} \sum_{i=1}^{N} \Delta(i,I) c_i z_i^F(t)$$

subject to

$$\left\{ z_i^I(t),\ z_i^F(t) \right\}_{i=1,\ldots,N}^{t=0,\ldots,T-1} \in A(L(T)),$$

where $L(T)$ consists of the constant histories

$$\left\{ \bar{z}_i \right\}_{i=1,\ldots,N},\ \left\{ X_k \right\}_{k=1,\ldots,NS},\ \left\{ Y_k \right\}_{k=NS+1,\ldots,NK},$$

$$\left\{ inv_i^0 \right\}_{i=1,\ldots,N},\ and \left\{ cap_i \right\}_{i=1,\ldots,N};$$

and the constraints $A(L(T))$ apply according to constant technical coefficients

$$\left\{ a_{ik} \right\}_{i=1,\ldots,N}^{k=1,\ldots,NK},\ \left\{ \bar{a}_{ij} \right\}_{i=1,\ldots,N}^{j=1,\ldots,N},\ and \left\{ c_i \right\}_{i=1,\ldots,N}.$$

Let

$$\left\{ z^{*I}_i(t),\ z^{*F}_i(t) \right\}_{i=1,\ldots,N}^{t=0,\ldots,T-1}$$

be optimal for the above linear program. One way of looking at a steady state model of production is to regard it as a statement of long run average relationships. For such a model one defines

$$(X_k,\ Y_k) \in R_+^{NK} \rightarrow \Phi(X_k,\ Y_k) \in R_+$$

by

$$\Phi(X_k,\ Y_k) = \lim_{T\to\infty} \frac{\sum\limits_{t=0}^{T-1} \sum\limits_{i=1}^{N} \Delta(i,l)c_i z^{*F}_i(t)}{T}. \tag{3.10}$$

This expression is a meaningful definition of the long run average output rate $\Phi(X_k,\ Y_k)$ if it can be shown that the limit exists.

We consider the change in output rate (total output divided by time horizon) for a unit increase in time horizon. Let

$$\left\{ z^{**I}_i(t),\ z^{**F}_i(t) \right\}_{i=1,\ldots,N}^{t=0,\ldots,T}$$

notate the optimal intensities for the output maximization program with time horizon $T+1$. Then

$$\begin{aligned}
&\bigg| \frac{1}{T+1} \sum_{t=0}^{T} \sum_{i=1}^{N} \Delta(i,1)c_i z^{**F}_i(t) - \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i=1}^{N} \Delta(i,1)c_i z^{*F}_i(t) \bigg| = \\
&\bigg| \frac{1}{T+1} \sum_{i=1}^{N} \Delta(i,1)c_i z^{**F}_i(T) + \frac{1}{T+1} \bigg[\sum_{t=0}^{T-1} \sum_{i=1}^{N} \Delta(i,1)c_i z^{**F}_i(t) - \\
&\sum_{t=0}^{T-1} \sum_{i=1}^{N} \Delta(i,1)c_i z^{*F}_i(t) - \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i=1}^{N} \Delta(i,1)c_i z^{*F}_i(t) \bigg] \bigg| \leqslant \\
&\frac{1}{T+1} \sum_{i=1}^{N} \Delta(i,1)c_i z^{**F}_i(T) + \frac{1}{T(T+1)} \sum_{t=0}^{T-1} \sum_{i=1}^{N} \Delta(i,1)c_i z^{*F}_i(t) + \\
&\frac{1}{T+1} \bigg| \sum_{t=0}^{T-1} \sum_{i=1}^{N} \Delta(i,1)c_i z^{**F}_i(t) - \sum_{t=0}^{T-1} \sum_{i=1}^{N} \Delta(i,1)c_i z^{*F}_i(t) \bigg|.
\end{aligned} \tag{3.11}$$

Without loss of generality, we may assume

$$z^{*I}_i(T-1) = 0,\ i=1,\ldots,N, \tag{3.12}$$

because these variables are constrained only to be non-negative by $\mathcal{A}(L(T))$.

and hence any optimal solution will remain optimal if these variables are set to zero. We claim that

$$\left\{z_i'^I(t),\ z_i'^F(t)\right\}_{i=1,\dots,N}^{t=0,\dots,T},$$

where

$$z_i'^I(t) = \begin{cases} z_i^{*I}(t) & \text{for } 0 \leqslant t \leqslant T-1 \\ 0 & t=T \end{cases} \quad , \tag{3.13}$$

and

$$z_i'^F(t) = \begin{cases} z_i^{*F}(t) & \text{for } 0 \leqslant t \leqslant T-1 \\ 0 & t=T \end{cases} \quad , \tag{3.14}$$

is feasible for the maximization program with time horizon $T+1$. Since $A(L(T+1))$ includes $A(L(T))$, it remains to investigate constraints for $t=T$. Resource constraints A(L(T))1, A(L(T))2 and intensity bound constraints A(L(T))5 are seen to be trivially true for $t=T$ due to zero intensities. Since

$$z_i^{*I}(T-1) = 0,\ i=1,\ \dots,N, \tag{3.15}$$

inventory constraints A(L(T))3 and A(L(T))4 are also satisfied.

Hence

$$\left\{z_i'^I(t),\ z_i'^F(t)\right\}_{i=1,\dots,N}^{t=0,\dots,T}$$

is indeed feasible for the program with horizon $T+1$, and by the optimality of

$$\left\{z_i^{**I}(t),\ z_i^{**F}(t)\right\}_{i=1,\dots,N}^{t=0,\dots,T},$$

we have

$$\sum_{t=0}^{T}\sum_{i=1}^{N} \Delta(i,1)c_i z_i'^F(t) = \sum_{t=0}^{T-1}\sum_{i=1}^{N} \Delta(i,1)c_i z_i^{*F}(t)$$

$$\leqslant \sum_{t=0}^{T}\sum_{i=1}^{N} \Delta(i,1)c_i z_i^{**F}(t). \tag{3.16}$$

Also,

$$\sum_{t=0}^{T-1}\sum_{i=1}^{N} \Delta(i,1)c_i z_i^{*F}(t) \leqslant \sum_{t=0}^{T-1}\sum_{i=1}^{N} \Delta(i,1)c_i z_i^{**F}(t) + \sum_{i=1}^{N} \Delta(i,1)c_i \Xi_i, \tag{3.17}$$

where the upper bounds on intensities have replaced $z_i^{**F}(T)$, $i=1,\ldots,N$. It is also the case that

$$\left\{ z_i^{**I}(t),\ z_i^{**F}(t) \right\}_{i=1,\ldots,N}^{t=0,\ldots,T-1} \in A(L(T)),$$

since these constraints are identical to those of $A(L(T+1))$ up to time $t=T-1$. By optimality of

$$\left\{ z_i^{*I}(t),\ z_i^{*F}(t) \right\}_{i=1,\ldots,N}^{t=0,\ldots,T-1},$$

we have

$$\sum_{t=0}^{T-1}\sum_{i=1}^{N} \Delta(i,1)c_i z_i^{**F}(t) \leqslant \sum_{t=0}^{T-1}\sum_{i=1}^{N} \Delta(i,1)c_i z_i^{*F}(t). \tag{3.18}$$

From (3.17) and (3.18), we have

$$\left| \sum_{t=0}^{T-1}\sum_{i=1}^{N} \Delta(i,1)c_i z_i^{*F}(t) - \sum_{t=0}^{T-1}\sum_{i=1}^{N} \Delta(i,1)c_i z_i^{**F}(t) \right| \leqslant \sum_{i=1}^{N} \Delta(i,1)c_i \Xi_i. \tag{3.19}$$

Substituting (3.19) into (3.11), we obtain

$$\left|\frac{1}{T+1}\sum_{t=0}^{T}\sum_{i=1}^{N}\Delta(i,1)c_i z_i^{**F}(t) - \frac{1}{T}\sum_{t=0}^{T-1}\sum_{i=1}^{N}\Delta(i,1)c_i z_i^{*F}(t)\right| \leqslant$$

$$\frac{1}{T+1}\sum_{i=1}^{N}\Delta(i,1)c_i z_i^{**F}(T) + \frac{1}{T(T+1)}\sum_{t=0}^{T-1}\sum_{i=1}^{N}\Delta(i,1)c_i z_i^{*F}(t) +$$

$$\frac{1}{T+1}\sum_{i=1}^{N}\Delta(i,1)c_i\Xi_i. \tag{3.20}$$

Replacing intensity variables by intensity bounds in the first two terms on the right hand side of (3.20), we have

$$\left|\frac{1}{T+1}\sum_{t=0}^{T}\sum_{i=1}^{N}\Delta(i,1)c_i z_i^{**F}(t) - \frac{1}{T}\sum_{t=0}^{T-1}\sum_{i=1}^{N}\Delta(i,1)c_i z_i^{*F}(t)\right| \leqslant$$

$$\frac{3}{T+1}\sum_{i=1}^{N}\Delta(i,1)c_i\Xi_i. \tag{3.21}$$

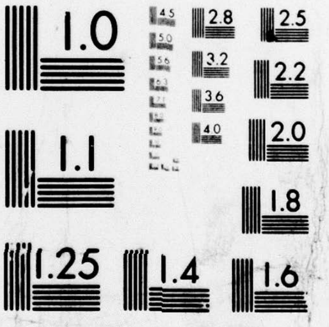
As T approaches infinity, the right hand side of (3.21) approaches zero. We conclude that there is a long run average output rate for a production system with exogenous input flows and technical data constant in time. A steady state model of production may thus be defined from the underlying dynamic structure.

In practice, this suggests an alternative to the typical econometric steady state models of production, in which statistical correlations are made of system input and output rates. Instead, an activity analysis could be performed, whereupon optimization of the output maximization program with a large time horizon would provide an estimate of the system's maximum output rate for given input rates.

AD-A074 901 CALIFORNIA UNIV BERKELEY OPERATIONS RESEARCH CENTER F/G 12/2
ASPECTS OF DYNAMIC PRODUCTION PLANNING.(U)
JUL 79 R C LEACHMAN N00014-76-C-0134
UNCLASSIFIED ORC-79-8 NL

2 OF 2
AD
A074901

END
DATE
FILMED
11-79
DDC



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

References

1. Berge, Claude, and Ghouila-Houri, A., *Programming, Games, and Transportation Networks,* John Wiley and Sons, New York (1965).

2. Gass, Saul I., *Linear Programming - Methods and Applications,* McGraw-Hill, New York (1975).

3. Herroloen, W., "Resource-Constrained Project Scheduling - The State of The Art," *Operational Research Quarterly,* Vol. 23, No. 3 (1972).

4. Lasdon, Leon S., *Optimization Theory For Large Systems,* MacMillan Co., New York (1970).

5. Moder, Joseph J., and Phillips, Cecil R., *Project Management With CPM and PERT,* Second Edition, Van Nostrand and Reinhold, New York (1970).

6. Patterson, James H., "Alternate Methods of Project Scheduling With Limited Resources," *Naval Logistics Quarterly,* Vol. 20, pg. 767-784 (1973).

7. Shephard, Ronald W., and Färe, Rolf, "Dynamic Theory of Production Correspondences, Part I," *Report ORC 78-2,* Operations Research Center, University of California, Berkeley (1978).

8. Shephard, Ronald W., and Färe, Rolf, "Dynamic Theory of Production Correspondences, Part II," *Report ORC 78-3,* Operations Research Center, University of California, Berkeley (1978).

9. Shephard, Ronald W., and Färe, Rolf, "Dynamic Theory of Production Correspondences, Part III," *Report ORC 78-4,* Operations Research Center, University of California, Berkeley (1978).

10. Shephard, Ronald W., and Al-Ayat, Rokaya, "Dynamic Theory of Production Correspondences, Part IV," *Report ORC 78-15,* Operations Research Center, University of California, Berkeley (1978).

11. Shephard, Ronald W., Al-Ayat, Rokaya, and Leachman, Robert C., "Shipbuilding Production Function: An Example of a Dynamic Production Function," *Report ORC 76-25,* Operations Research Center, University of California, Berkeley (1976).

12. Weist, Jerome D., "A Heuristic Model for Scheduling Large Projects With Limited Resources," *Management Science,* Vol. 13, No. 6 (1967).